

# INVITATION

## Conference Chair: Mamdouh Ibrahim, IBM Global Services



"Don't miss the premier forum for sharing of knowledge on object technology."

On behalf of the OOPSLA 2002 Conference Committee, I would like to invite you to participate in this year's OOPSLA, which will be held in Seattle, Washington, 4-8 November 2002.

OOPSLA is the premier forum for bringing together practitioners, researchers, and students to share their ideas and experiences in a broad range

of disciplines woven with the common thread of object technology. Over two hundred people (mostly volunteers) have done their best to create an outstanding technical conference with a wide variety of events and activities designed to benefit a wide spectrum of participants ranging from novices to experts and everyone in between.

OOPSLA 2002 continues the tradition established by previous OOPSLA conferences to provide carefully refereed technical papers, real world experiences in the form of practitioner reports, and exciting panels and debates. This year, OOPSLA 2002 will feature outstanding invited speakers sharing their vision and delivering thought-provoking topics including Bill Gates, Kent Beck, Alfred Spector, Jerry Michalski, Anders Hejlsberg, and for the first time the Turing Lecture will be presented at OOPSLA by Kris Nygaard the winner of the Turing Award 2001.

New this year are two special tracks that promise to be of interest to both researchers and practitioners. The Onward! Track is designed to foster thinking out of the box in search for new computing models, and the focus of the second track is the hot topic of Web Services.

The OOPSLA conferences are known for their excellent tutorials and highly interactive, topic-focused workshops; this year is no different. I would strongly recommend you register for tutorials of your choice as soon as possible since they tend to sell out fast. I would also encourage you to participate in one or more workshops. It is easier than you might imagine to be invited to a workshop; so if you have never participated in a workshop before try this year and I am confident you will find the experience very rewarding.

You may also wish to participate in some of the more specialized activities—the Doctoral Symposium for Ph.D. students to get feedback on their research ideas, the Educators' Symposium for academic and industrial educators to share new ideas and best practices, as well as our ever-popular DesignFest and CodeFest. You can also organize your own Birds of a Feather session after arriving at the conference. Also, this year the OOPSLA Poster Session has been extended to include the first ACM SIGPLAN Student Research Competition, which was made possible by the support of Microsoft Research.

Based on last year's popularity, the OOPSLA Courtyard will once again be the hub of ongoing interactive activities during the three main conference days (Wednesday-Friday). In the Courtyard you will find commercial exhibits of the latest products, news, and publications. Also, the exhibit will feature the successful "geek alley" where you can pick up your favorite personal devices and software, and see posters and demonstrations that present late-breaking results (including summaries of some pre-conference events), speaker follow-up sessions, Birds of a Feather sign-ups, job and message boards, and tables for impromptu demos or discussions. And of course, we have plenty of social opportunities for mingling and professional networking. In fact, OOPSLA networking opportunities are unparalleled!

Co-resident in Seattle with OOPSLA 2002 is JavaGrande. Careful consideration of scheduling allows you to participate in both if you are interested.

OOPSLA has so much to offer that you will have to make choices. So study the Advance Program carefully as you plan your OOPSLA experience. We have arranged a special orientation session immediately following the Welcome Reception for first-time attendees.

We look forward to seeing you in Seattle!

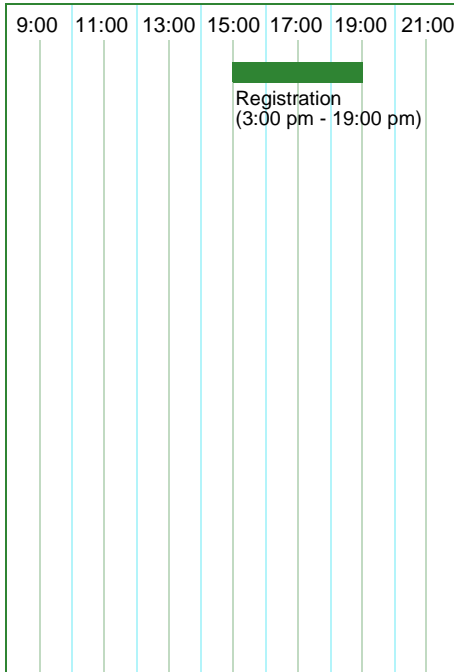
---

## Table of Contents

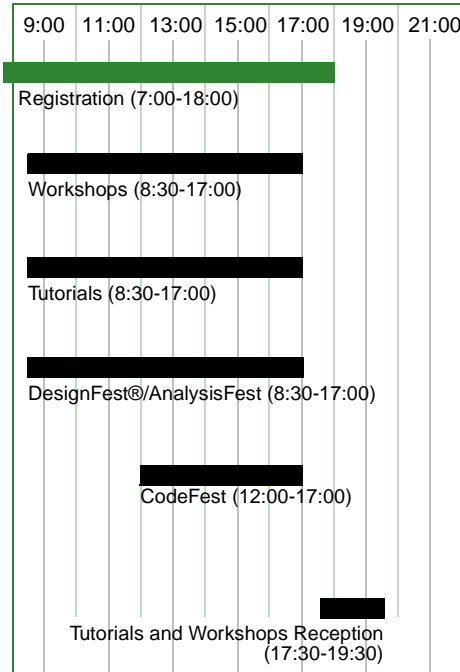
Conference At A Glance .....	3
Tracks .....	7
Technical Program .....	18
Tutorials .....	45
Workshops .....	77
DesignFest .....	89
Educators' Symposium .....	90
Doctoral Symposium .....	92
Demonstrations .....	92
Posters .....	93
The Joint ACM SIGPLAN Student Research Competition and OOPSLA Poster Session .....	93
Student Volunteers .....	94
Special Events .....	95
Birds of a Feather Sessions (BOFs) .....	96
Camp Smalltalk .....	96
Exhibits .....	97
Housing information .....	99
Housing Scholarship for Qualified Academic Faculty .....	101
Registration Information .....	102
Transportation Information .....	106
ACM / SIGPLAN / SIGSOFT .....	106
Seattle Facts .....	107
OOPSLA 2002 Committees .....	108

## OOPSLA 2002 AT A GLANCE

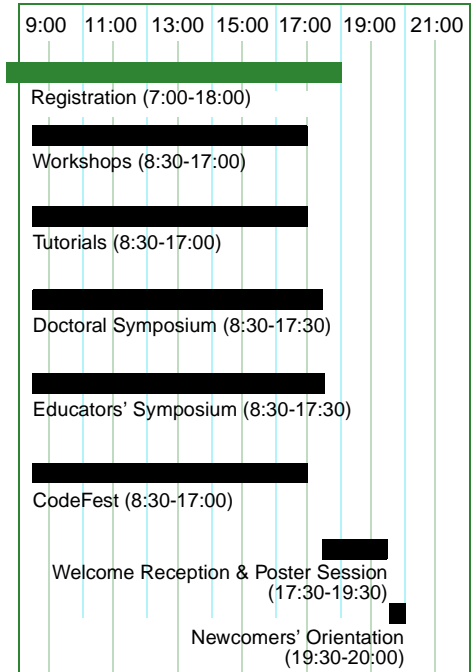
## Sunday



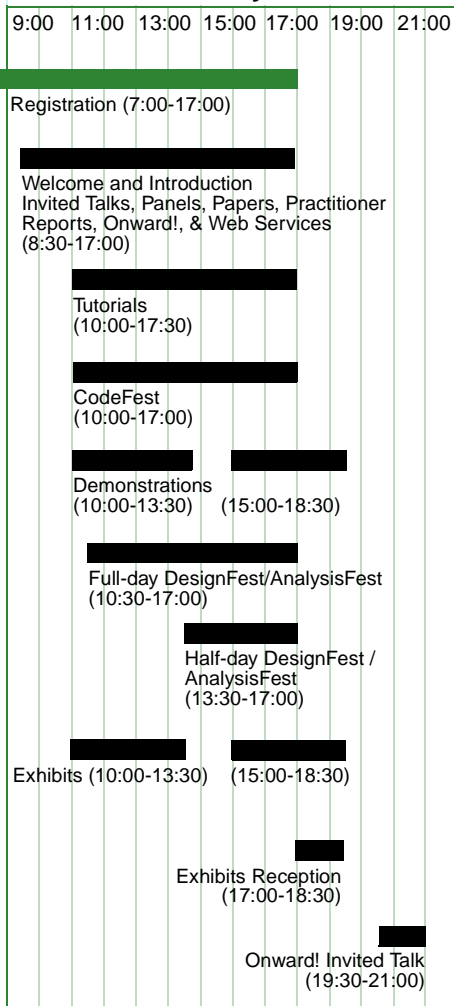
## Monday



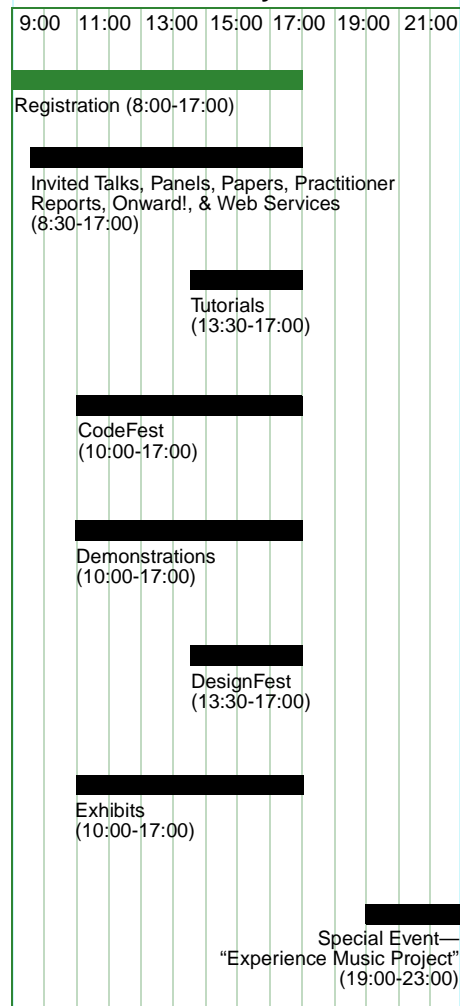
## Tuesday



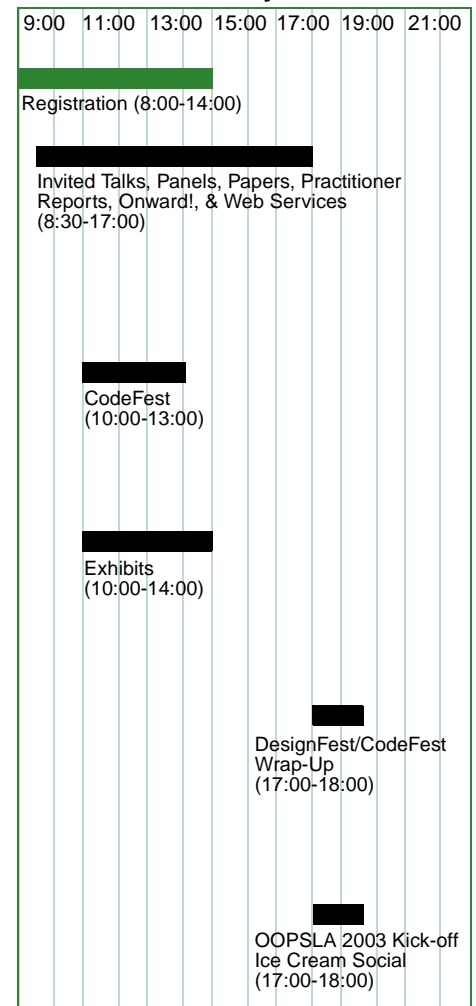
## Wednesday



## Thursday



## Friday



# CONFERENCE AT A GLANCE

## Monday, 4 November

### 8:30-17:00

- Tutorial 1: Essential Object-oriented Analysis and Design, *page 47*
- Tutorial 2: Agile Software Development Methodologies: Principles and Practices, *page 47*
- Tutorial 3: Pattern-Oriented Software Architectures for Networked and Concurrent Applications, *page 48*
- Tutorial 4: Automated Software Testing: Hands On and Interactive!, *page 48*
- Tutorial 5: Pragmatic Programming - the First Day, *page 49*
- Tutorial 6: Programming Web Services Applications, *page 50*
- Workshop 1: Tool Support for Aspect Oriented Software Development, *page 78*
- Workshop 2: 2nd OOPSLA Workshop on Domain-Specific Visual Languages, *page 78*
- Workshop 3: Commonalities of Agile Methodologies, *page 79*
- Workshop 7: Top 10 issues of an open software model for embedded systems, *page 79*
- Workshop 8: The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- Workshop 9: Pervasive Computing: going beyond Internet for small screens, *page 80*
- Workshop 10: Distributed eXtreme Programming, *page 81*
- Workshop 12: Patterns for Software Architecture, *page 81*
- Workshop 14: Agent-oriented methodologies, *page 81*
- Workshop 15: Tackling the Discovery Costs of Evolving Software Systems, *page 81*
- Workshop 22: J2EE vs .NET, *page 82*
- Workshop 23: Using Play to Enhance Learning About Objects, *page 82*
- Workshop 25: "Killer Examples" for Design Patterns and Objects First, *page 82*
- DesignFest: Full Day DesignFest, *page 89*

### 8:30-12:00

- Tutorial 7: Building Secure OO Systems - A Practical Guide and Overview, *page 51*
- Tutorial 8: Introduction to the .NET Framework, *page 51*
- Tutorial 9: The Art of Writing Use Cases, *page 52*
- Tutorial 10: Programmer's Dozen: Thirteen Practices for Refactoring, Repairing, and Regaining Control of Your Code, *page 53*
- Tutorial 11: Efficient Implementation of Object-Oriented Programming Languages, *page 53*
- Workshop 5: Extreme Programming Practices in the First CS Courses, *page 83*

### 12:00-17:00

- CodeFest, *page 89*

### 13:30-17:00

- Tutorial 12: Card-based User and Use Case Task Modeling, *page 54*
- Tutorial 13: Developing Java Applications for Mobile Devices, *page 54*
- Tutorial 14: Object vs. The Web, *page 55*
- Tutorial 15: Patterns of Enterprise Application Architecture, *page 55*
- Tutorial 16: Aspect-Oriented Programming in C++, *page 56*
- Tutorial 17: The Design and Implementation of the Jikes RVM Optimizing Compiler, *page 56*
- Workshop 31: Expanding the Boundaries of Unit Testing, *page 83*

### 17:30-19:30

- Special Event: Tutorial and Workshop Reception, *page 95*

## Tuesday, 5 November

### 8:30-17:30

---

- Educators' Symposium, *page 90*
- Doctoral Symposium, *page 92*

### 8:30-17:00

---

- Tutorial 18: A Brief Tour of Responsibility-Driven Design, *page 57*
- Tutorial 19: Concepts of Object-Oriented Programming, *page 58*
- Tutorial 20: Web Services and Service Oriented Architectures, *page 58*
- Tutorial 21: A Language Designer's View of Rotor, Microsoft's "Shared Source" Implementation of the Common Language Infrastructure, *page 59*
- Workshop 4: Generative Techniques in the context of Model-Driven Architecture, *page 84*
- Workshop 6: Architecture is dead - Long live the Architect!, *page 84*
- Workshop 11: Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- Workshop 13: Patterns for Customer Interaction and Expectation Management, *page 85*
- Workshop 16: 1st Workshop on Software Development Process Patterns, *page 85*
- Workshop 17: Built for Life: Constructing Software to Outlive Its Creators, *page 85*
- Workshop 18: Semantics of Enterprise Integration II, *page 86*
- Workshop 19: Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE), *page 86*
- Workshop 20: Object-Oriented Web Services, *page 87*
- Workshop 21: Software Apprenticeship: the Journey to Mastery, *page 87*
- Workshop 24: 1st International Workshop on Runtime Kernel Support for Dynamic Languages and Component Based Architectures, *page 82*
- Workshop 26: Using Domain Specific Languages to Drive Business Applications, *page 87*
- Workshop 27: Extravagaria - Art Assisting Science, *page 87*
- Workshop 28: Agile Processes Workshop II - Managing Multiple Agile Projects, *page 87*
- Workshop 30: Product Line Engineering - The early steps: Planning, Managing, and Modeling (PLEES'02), *page 88*
- CodeFest, *page 89*

### 8:30-12:00

---

- Tutorial 22: Extreme Programming, A Simulation, *page 60*
- Tutorial 23: Designing an Agile Methodology, *page 60*
- Tutorial 24: J2EE for Enterprise Application Integration and e-business Integration, *page 61*
- Tutorial 25: Introduction to Concurrent Programming in Java, *page 61*
- Tutorial 26: Reflection in Java, *page 62*
- Tutorial 27: Patterns at Work, *page 62*
- Tutorial 28: Use Cases, Usability Requirements and User Interfaces, *page 63*
- Tutorial 29: .NET Primer with C#, *page 63*
- Workshop 29: Pair Programming Explored, *page 88*

### 13:30-17:00

---

- Tutorial 30: eXtreme Programming & Web Development, *page 64*
- Tutorial 31: Patterns for Writing Effective Use Cases, *page 64*
- Tutorial 32: Dungeons and Patterns!, *page 65*
- Tutorial 33: Concurrency Utilities: Tools for Managing Multithreading, *page 65*
- Tutorial 34: C++ Idioms, *page 66*
- Tutorial 35: What's Happening Inside Your EJB Application Server?, *page 66*
- Tutorial 36: Patterns and Application Experiences for Real-time Object Request Brokers, *page 67*
- Tutorial 37: Objects, XML and Databases, *page 67*
- Tutorial 38: Distributed .NET, *page 68*

### 17:30-19:30

---

- Special Event: OOPSLA 2002 Welcome Reception and Poster Session, *page 95*

### 19:30-20:00

---

- Special Event: Newcomers Orientation, *page 95*

## Wednesday, 6 November

**8:30-10:00**

- Invited Speaker: The Metaphor Metaphor, *page 28*

**10:00-17:30**

- Tutorial 39: Ruby in a Day, *page 69*
- Tutorial 40: Aspect-Oriented Programming with AspectJ, *page 69*
- Tutorial 41: Programming Web Services for Mobile Devices, *page 70*

**10:00-17:00**

- CodeFest, *page 89*

**10:00-13:30**

- Demonstrations, *page 92*
- Exhibit: Exhibits Open

**10:00-13:15**

- Tutorial 42: Refactoring To Patterns, *page 71*
- Tutorial 43: EJB Roleplay, *page 71*
- Tutorial 44: Test-Driven Development in .NET, *page 71*

**10:30-17:00**

- DesignFest: Full Day DesignFest, *page 89*

**10:30-12:00**

- Technical Papers: Storage Management, *page 29*
- Panel: Resolved: Objects Have Failed (Part 1 of 2), *page 30*
- Practitioner Reports: Being Agile, *page 30*

**13:30-17:00**

- Tutorial 45: Object-Oriented Reengineering: Patterns & Techniques, *page 72*
- Tutorial 46: Patterns for EJB Development, *page 72*
- Tutorial 47: Scrum and Agile Process 101, *page 73*
- Tutorial 48: Use Cases and Testing: Using Use Cases to Write Test Cases, *page 73*
- DesignFest: Half Day DesignFest, *page 89*

**13:30-15:00**

- Technical Papers: Components and Composition, *page 31*
- Panel: The Emperor's New Design, *page 31*
- Onward! Papers: New Models for Software I, *page 31*

**15:00-18:30**

- Demonstrations, *page 92*
- Exhibit: Exhibits Open

**15:30-17:00**

- Technical Papers: Languages, *page 32*
- Web Services Panel: If I Had a Hammer: Viewing Web Services as Your Favorite Nail, *page 32*
- Onward! Panel: Biologically Inspired Software, *page 32*

**17:00-18:30**

- Special Event: Exhibitors Reception, *page 95*

**19:30-21:00**

- Onward! Invited Speaker: What's Next?, *page 33*

## Thursday, 7 November

## 8:30-10:00

- Turing Lecture: The Development of the Key Object-Oriented Concepts, *page 34*

## 10:00-17:00

- CodeFest, *page 89*
- Demonstrations, *page 92*
- Exhibit: Exhibits Open

## 10:30-12:00

- Technical Papers: Optimizations, *page 34*
- Web Services Panel: Postcards from the Bleeding Edge: Web Service User/Developer Experiences, *page 35*
- Onward! Papers: New Models for Software II, *page 35*

## 13:30-17:00

- Tutorial 49: How to Use Design Patterns In Java and .NET, *page 74*
- Tutorial 50: Daily Builds are for Wimps, *page 74*
- Tutorial 51: Successful Technical Leading on an OO project, *page 75*
- Tutorial 52: Notes on the forgotten Art of Software Architecture, *page 75*
- Tutorial 53: Framework Design and Implementation using Java and UML, *page 76*
- Tutorial 54: Developing Web Services, *page 76*
- DesignFest: Half Day DesignFest, *page 89*

## 13:30-15:00

- Invited Speaker: The C# Programming Language, *page 35*
- Technical Papers: Aspects, *page 35*
- Onward! Papers: New Programming Constructs, *page 36*

## 15:30-17:00

- Technical Papers: Static Analysis, *page 36*
- Panel: To Be Extreme, or Not to Be Extreme, *page 37*
- Practitioner Reports: In the Trenches: Tough Problems, *page 37*

## 19:00-23:00

- Special Event: Evening at “Experience Music Project”, *page 95*

## Friday, 8 November

## 8:30-10:00

- Web Services Invited Speaker: Delivering on the promise of distributed systems, *page 38*
- Panel: Object-Oriented Success Stories, *page 38*

## 10:00-14:00

- Exhibit: Exhibits Open

## 10:00-13:00

- CodeFest, *page 89*

## 10:30-12:00

- Technical Papers: Scalability, *page 39*
- Technical Papers: Foundations, *page 40*
- Practitioner Reports: Architecture Experiences, *page 40*
- Panel: The Failure of Object Education: What went wrong? What can we do differently?, *page 41*

## 12:00-13:00

- Onward! Panel: New Programming Constructs Beyond Inheritance, Patterns, and Notation: What’s left?, *page 41*

## 13:30-15:00

- Web Services Invited Speaker: The Future of Programming in a World of Web Services, *page 42*

## 15:30-17:00

- Technical Papers: Tools, *page 43*
- Panel: Objects and Real Time: Collision or Spiritual Union, *page 44*
- Panel: Resolved: Objects Have Failed (Part 2 of 2), *page 44*

## 17:00-18:00

- DesignFest: DesignFest/CodeFest Wrap-Up, *page 89*
- Special Event: OOPSLA 2003 Kick-off (Ice Cream Social), *page 95*

# TRACKS

## METHODS & PROCESSES

### Monday, 4 November

#### 8:30-17:00

- Tutorial 1: Essential Object-oriented Analysis and Design, *page 47*
- Workshop 1: Tool Support for Aspect Oriented Software Development, *page 78*
- Workshop 2: 2nd OOPSLA Workshop on Domain-Specific Visual Languages, *page 78*
- Workshop 3: Commonalities of Agile Methodologies, *page 79*
- Workshop 8: The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- Workshop 10: Distributed eXtreme Programming, *page 81*
- Workshop 14: Agent-oriented methodologies, *page 81*
- Workshop 15: Tackling the Discovery Costs of Evolving Software Systems, *page 81*

#### 8:30-12:00

- Tutorial 9: The Art of Writing Use Cases, *page 52*
- Workshop 5: Extreme Programming Practices in the First CS Courses, *page 83*

#### 13:30-17:00

- Tutorial 12: Card-based User and Use Case Task Modeling, *page 54*

### Tuesday, 5 November

#### 8:30-17:00

- Tutorial 18: A Brief Tour of Responsibility-Driven Design, *page 57*
- Workshop 4: Generative Techniques in the context of Model-Driven Architecture, *page 84*
- Workshop 6: Architecture is dead - Long live the Architect!, *page 84*
- Workshop 16: 1st Workshop on Software Development Process Patterns, *page 85*
- Workshop 17: Built for Life: Constructing Software to Outlive Its Creators, *page 85*
- Workshop 21: Software Apprenticeship: the Journey to Mastery, *page 87*
- Workshop 30: Product Line Engineering - The early steps: Planning, Managing, and Modeling (PLEES'02), *page 88*

### Tuesday (cont.)

#### 8:30-12:00

- Tutorial 22: Extreme Programming, A Simulation, *page 60*
- Tutorial 23: Designing an Agile Methodology, *page 60*
- Tutorial 24: J2EE for Enterprise Application Integration and e-business Integration, *page 61*
- Tutorial 28: Use Cases, Usability Requirements and User Interfaces, *page 63*
- Workshop 29: Pair Programming Explored, *page 88*

#### 13:30-17:00

- Tutorial 30: eXtreme Programming & Web Development, *page 64*
- Tutorial 31: Patterns for Writing Effective Use Cases, *page 64*

### Wednesday, 6 November

#### 10:30-12:00

- Practitioner Reports: Being Agile, *page 30*

#### 13:30-17:00

- Tutorial 45: Object-Oriented Reengineering: Patterns & Techniques, *page 72*
- Tutorial 47: Scrum and Agile Process 101, *page 73*
- Tutorial 48: Use Cases and Testing: Using Use Cases to Write Test Cases, *page 73*

### Thursday, 7 November

#### 13:30-17:00

- Tutorial 51: Successful Technical Leading on an OO project, *page 75*
- Tutorial 52: Notes on the forgotten Art of Software Architecture, *page 75*

#### 15:30-17:00

- Panel: To Be Extreme, or Not to Be Extreme, *page 37*



## LANGUAGES

### Monday, 4 November

#### 8:30-17:00

- Workshop 1: Tool Support for Aspect Oriented Software Development, *page 78*
- Workshop 2: 2nd OOPSLA Workshop on Domain-Specific Visual Languages, *page 78*
- Workshop 22: J2EE vs .NET, *page 82*

#### 8:30-12:00

- Tutorial 11: Efficient Implementation of Object-Oriented Programming Languages, *page 53*

#### 13:30-17:00

- Tutorial 13: Developing Java Applications for Mobile Devices, *page 54*
- Tutorial 16: Aspect-Oriented Programming in C++, *page 56*
- Tutorial 17: The Design and Implementation of the Jikes RVM Optimizing Compiler, *page 56*

### Tuesday, 5 November

#### 8:30-17:00

- Tutorial 19: Concepts of Object-Oriented Programming, *page 58*
- Tutorial 21: A Language Designer's View of Rotor, Microsoft's "Shared Source" Implementation of the Common Language Infrastructure, *page 59*
- Workshop 4: Generative Techniques in the context of Model-Driven Architecture, *page 84*
- Workshop 24: 1st International Workshop on Runtime Kernel Support for Dynamic Languages and Component Based Architectures, *page 82*
- Workshop 26: Using Domain Specific Languages to Drive Business Applications, *page 87*

#### 8:30-12:00

- Tutorial 26: Reflection in Java, *page 62*

#### 13:30-17:00

- Tutorial 34: C++ Idioms, *page 66*

### Wednesday, 6 November

#### 10:00-17:30

- Tutorial 39: Ruby in a Day, *page 69*
- Tutorial 40: Aspect-Oriented Programming with AspectJ, *page 69*

#### 10:00-13:15

- Tutorial 43: EJB Roleplay, *page 71*

#### 13:30-17:00

- Tutorial 46: Patterns for EJB Development, *page 72*

#### 13:30-15:00

- Technical Papers: Components and Composition, *page 31*

#### 15:30-17:00

- Technical Papers: Languages, *page 32*

### Thursday, 7 November

#### 8:30-10:00

- Turing Lecture: The Development of the Key Object-Oriented Concepts, *page 34*

#### 10:30-12:00

- Technical Papers: Optimizations, *page 34*

#### 13:30-17:00

- Tutorial 49: How to Use Design Patterns In Java and .NET, *page 74*

#### 13:30-15:00

- Invited Speaker: The C# Programming Language, *page 35*
- Technical Papers: Aspects, *page 35*

#### 15:30-17:00

- Technical Papers: Static Analysis, *page 36*

### Friday, 8 November

#### 10:30-12:00

- Technical Papers: Scalability, *page 39*
- Technical Papers: Foundations, *page 40*

#### 15:30-17:00

- Technical Papers: Tools, *page 43*



## AGILE

### Monday, 4 November

#### 8:30-17:00

- Tutorial 2: Agile Software Development Methodologies: Principles and Practices, *page 47*
- Tutorial 4: Automated Software Testing: Hands On and Interactive!, *page 48*
- Workshop 3: Commonalities of Agile Methodologies, *page 79*
- Workshop 10: Distributed eXtreme Programming, *page 81*
- DesignFest: Full Day DesignFest, *page 89*

#### 8:30-12:00

- Workshop 5: Extreme Programming Practices in the First CS Courses, *page 83*

#### 13:30-17:00

- Workshop 31: Expanding the Boundaries of Unit Testing, *page 83*

### Tuesday, 5 November

#### 8:30-17:00

- Workshop 6: Architecture is dead - Long live the Architect!, *page 84*
- Workshop 16: 1st Workshop on Software Development Process Patterns, *page 85*
- Workshop 28: Agile Processes Workshop II - Managing Multiple Agile Projects, *page 87*

#### 8:30-12:00

- Tutorial 22: Extreme Programming, A Simulation, *page 60*
- Tutorial 23: Designing an Agile Methodology, *page 60*

#### 13:30-17:00

- Tutorial 30: eXtreme Programming & Web Development, *page 64*

### Wednesday, 6 November

#### 8:30-10:00

- Invited Speaker: The Metaphor Metaphor, *page 28*

#### 10:00-13:15

- Tutorial 42: Refactoring To Patterns, *page 71*
- Tutorial 44: Test-Driven Development in .NET, *page 71*

#### 10:30-17:00

- DesignFest: Full Day DesignFest, *page 89*

#### 10:30-12:00

- Practitioner Reports: Being Agile, *page 30*

#### 13:30-17:00

- Tutorial 47: Scrum and Agile Process 101, *page 73*
- DesignFest: Half Day DesignFest, *page 89*

### Thursday, 7 November

#### 13:30-17:00

- Tutorial 50: Daily Builds are for Wimps, *page 74*
- DesignFest: Half Day DesignFest, *page 89*

#### 15:30-17:00

- Panel: To Be Extreme, or Not to Be Extreme, *page 37*
- Practitioner Reports: In the Trenches: Tough Problems, *page 37*

### Friday, 8 November

#### 10:30-12:00

- Practitioner Reports: Architecture Experiences, *page 40*

## TECHNOLOGIES

### Monday, 4 November

#### 8:30-17:00

- Tutorial 3: Pattern-Oriented Software Architectures for Networked and Concurrent Applications, *page 48*
- Workshop 1: Tool Support for Aspect Oriented Software Development, *page 78*
- Workshop 2: 2nd OOPSLA Workshop on Domain-Specific Visual Languages, *page 78*
- Workshop 7: Top 10 issues of an open software model for embedded systems, *page 79*
- Workshop 9: Pervasive Computing; going beyond Internet for small screens, *page 80*
- Workshop 22: J2EE vs .NET, *page 82*

#### 8:30-12:00

- Tutorial 8: Introduction to the .NET Framework, *page 51*

#### 13:30-17:00

- Tutorial 13: Developing Java Applications for Mobile Devices, *page 54*
- Tutorial 14: Object vs. The Web, *page 55*

### Tuesday, 5 November

#### 8:30-17:00

- Workshop 11: Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- Workshop 18: Semantics of Enterprise Integration II, *page 86*
- Workshop 19: Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE), *page 86*
- Workshop 30: Product Line Engineering - The early steps: Planning, Managing, and Modeling (PLEES'02), *page 88*

#### 8:30-12:00

- Tutorial 24: J2EE for Enterprise Application Integration and e-business Integration, *page 61*
- Tutorial 25: Introduction to Concurrent Programming in Java, *page 61*
- Workshop 29: Pair Programming Explored, *page 88*

#### 13:30-17:00

- Tutorial 33: Concurrency Utilities: Tools for Managing Multithreading, *page 65*
- Tutorial 35: What's Happening Inside Your EJB Application Server?, *page 66*
- Tutorial 36: Patterns and Application Experiences for Real-time Object Request Brokers, *page 67*
- Tutorial 37: Objects, XML and Databases, *page 67*

### Wednesday, 6 November

#### 10:00-17:30

- Tutorial 41: Programming Web Services for Mobile Devices, *page 70*

#### 10:30-12:00

- Technical Papers: Storage Management, *page 29*

#### 13:30-15:00

- Technical Papers: Components and Composition, *page 31*

#### 15:30-17:00

- Technical Papers: Languages, *page 32*

### Thursday, 7 November

#### 10:30-12:00

- Technical Papers: Optimizations, *page 34*

#### 13:30-15:00

- Technical Papers: Aspects, *page 35*

#### 15:30-17:00

- Technical Papers: Static Analysis, *page 36*

### Friday, 8 November

#### 10:30-12:00

- Technical Papers: Scalability, *page 39*

#### 15:30-17:00

- Technical Papers: Tools, *page 43*

## WEB SERVICES

### Monday, 4 November

#### 8:30-17:00

- Tutorial 6: Programming Web Services Applications, *page 50*
- Workshop 22: J2EE vs .NET, *page 82*

#### 8:30-12:00

- Tutorial 8: Introduction to the .NET Framework, *page 51*

### Tuesday, 5 November

#### 8:30-17:00

- Tutorial 20: Web Services and Service Oriented Architectures, *page 58*
- Workshop 20: Object-Oriented Web Services, *page 87*

#### 8:30-12:00

- Tutorial 29: .NET Primer with C#, *page 63*

#### 13:30-17:00

- Tutorial 38: Distributed .NET, *page 68*

### Wednesday, 6 November

#### 10:00-17:30

- Tutorial 41: Programming Web Services for Mobile Devices, *page 70*

#### 10:00-13:15

- Tutorial 44: Test-Driven Development in .NET, *page 71*

#### 15:30-17:00

- Web Services Panel: If I Had a Hammer: Viewing Web Services as Your Favorite Nail, *page 32*

### Thursday, 7 November

#### 10:30-12:00

- Web Services Panel: Postcards from the Bleeding Edge: Web Service User/Developer Experiences, *page 35*

#### 13:30-17:00

- Tutorial 50: Daily Builds are for Wimps, *page 74*
- Tutorial 54: Developing Web Services, *page 76*

### Friday, 8 November

#### 8:30-10:00

- Web Services Invited Speaker: Delivering on the promise of distributed systems, *page 38*

#### 10:30-12:00

- Practitioner Reports: Architecture Experiences, *page 40*
- Technical Papers: Scalability, *page 39*

#### 13:30-15:00

- Web Services Invited Speaker: The Future of Programming in a World of Web Services, *page 42*

## ANALYSIS, DESIGN & ARCHITECTURE

### Monday, 4 November

#### 8:30-17:00

- Tutorial 1: Essential Object-oriented Analysis and Design, *page 47*
- Workshop 2: 2nd OOPSLA Workshop on Domain-Specific Visual Languages, *page 78*
- Workshop 7: Top 10 issues of an open software model for embedded systems, *page 79*
- Workshop 8: The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- Workshop 9: Pervasive Computing; going beyond Internet for small screens, *page 80*
- Workshop 12: Patterns for Software Architecture, *page 81*
- Workshop 25: “Killer Examples” for Design Patterns and Objects First, *page 82*
- DesignFest: Full Day DesignFest, *page 89*

#### 8:30-12:00

- Tutorial 7: Building Secure OO Systems - A Practical Guide and Overview, *page 51*
- Workshop 5: Extreme Programming Practices in the First CS Courses, *page 83*

#### 12:00-17:00

- CodeFest, *page 89*

#### 13:30-17:00

- Tutorial 12: Card-based User and Use Case Task Modeling, *page 54*
- Tutorial 14: Object vs. The Web, *page 55*
- Tutorial 15: Patterns of Enterprise Application Architecture, *page 55*

### Tuesday, 5 November

#### 8:30-17:00

- Tutorial 18: A Brief Tour of Responsibility-Driven Design, *page 57*
- Workshop 4: Generative Techniques in the context of Model-Driven Architecture, *page 84*
- Workshop 11: Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- Workshop 17: Built for Life: Constructing Software to Outlive Its Creators, *page 85*
- Workshop 26: Using Domain Specific Languages to Drive Business Applications, *page 87*
- Workshop 30: Product Line Engineering - The early steps: Planning, Managing, and Modeling (PLEES'02), *page 88*
- CodeFest, *page 89*

#### 8:30-12:00

- Tutorial 24: J2EE for Enterprise Application Integration and e-business Integration, *page 61*
- Tutorial 27: Patterns at Work, *page 62*

#### 13:30-17:00

- Tutorial 32: Dungeons and Patterns!, *page 65*

### Wednesday, 6 November

#### 10:00-17:00

- CodeFest, *page 89*

#### 10:30-17:00

- DesignFest: Full Day DesignFest, *page 89*

#### 13:30-17:00

- Tutorial 45: Object-Oriented Reengineering: Patterns & Techniques, *page 72*
- Tutorial 46: Patterns for EJB Development, *page 72*
- DesignFest: Half Day DesignFest, *page 89*

#### 13:30-15:00

- Technical Papers: Components and Composition, *page 31*
- Panel: The Emperor's New Design, *page 31*

## ANALYSIS, DESIGN & ARCHITECTURE (CONT.)

### Thursday, 7 November

#### 10:00-17:00

- CodeFest, *page 89*

#### 13:30-17:00

- Tutorial 49: How to Use Design Patterns In Java and .NET, *page 74*
- Tutorial 51: Successful Technical Leading on an OO project, *page 75*
- Tutorial 52: Notes on the forgotten Art of Software Architecture, *page 75*
- Tutorial 53: Framework Design and Implementation using Java and UML, *page 76*
- DesignFest: Half Day DesignFest, *page 89*

#### 15:30-17:00

- Technical Papers: Static Analysis, *page 36*
- Practitioner Reports: In the Trenches: Tough Problems, *page 37*

### Friday, 8 November

#### 10:00-13:00

- CodeFest, *page 89*

#### 10:30-12:00

- Technical Papers: Foundations, *page 40*
- Practitioner Reports: Architecture Experiences, *page 40*

#### 17:00-18:00

- DesignFest: DesignFest/CodeFest Wrap-Up, *page 89*

## INTRIGUING

### Monday, 4 November

#### 8:30-17:00

- Workshop 7: Top 10 issues of an open software model for embedded systems, *page 79*
- Workshop 8: The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- Workshop 10: Distributed eXtreme Programming, *page 81*
- Workshop 22: J2EE vs .NET, *page 82*

### Tuesday, 5 November

#### 8:30-17:00

- Workshop 11: Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- Workshop 26: Using Domain Specific Languages to Drive Business Applications, *page 87*
- Workshop 27: Extravagaria - Art Assisting Science, *page 87*

### Wednesday, 6 November

#### 13:30-15:00

- Onward! Papers: New Models for Software I, *page 31*

#### 15:30-17:00

- Onward! Panel: Biologically Inspired Software, *page 32*

#### 19:30-21:00

- Onward! Invited Speaker: What's Next?, *page 33*

### Thursday, 7 November

#### 10:30-12:00

- Onward! Papers: New Models for Software II, *page 35*

#### 13:30-15:00

- Onward! Papers: New Programming Constructs, *page 36*

### Friday, 8 November

#### 12:00-13:00

- Onward! Panel: New Programming Constructs Beyond Inheritance, Patterns, and Notation: What's left?, *page 41*

## PROGRAMMING

### Monday, 4 November

#### 8:30-17:00

- Tutorial 4: Automated Software Testing: Hands On and Interactive!, *page 48*
- Tutorial 5: Pragmatic Programming - the First Day, *page 49*
- Workshop 1: Tool Support for Aspect Oriented Software Development, *page 78*
- Workshop 7: Top 10 issues of an open software model for embedded systems, *page 79*
- Workshop 9: Pervasive Computing: going beyond Internet for small screens, *page 80*
- Workshop 10: Distributed eXtreme Programming, *page 81*
- Workshop 22: J2EE vs .NET, *page 82*
- Workshop 25: “Killer Examples” for Design Patterns and Objects First, *page 82*

#### 8:30-12:00

- Tutorial 7: Building Secure OO Systems - A Practical Guide and Overview, *page 51*
- Tutorial 8: Introduction to the .NET Framework, *page 51*
- Tutorial 10: Programmer’s Dozen: Thirteen Practices for Refactoring, Repairing, and Regaining Control of Your Code, *page 53*

#### 12:00-17:00

- CodeFest, *page 89*

#### 13:30-17:00

- Tutorial 13: Developing Java Applications for Mobile Devices, *page 54*
- Tutorial 14: Object vs. The Web, *page 55*
- Tutorial 16: Aspect-Oriented Programming in C++, *page 56*

### Tuesday, 5 November

#### 8:30-17:00

- Tutorial 19: Concepts of Object-Oriented Programming, *page 58*
- Workshop 11: Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- Workshop 17: Built for Life: Constructing Software to Outlive Its Creators, *page 85*
- Workshop 19: Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE), *page 86*
- Workshop 21: Software Apprenticeship: the Journey to Mastery, *page 87*
- Workshop 24: 1st International Workshop on Runtime Kernel Support for Dynamic Languages and Component Based Architectures, *page 82*
- Workshop 26: Using Domain Specific Languages to Drive Business Applications, *page 87*
- CodeFest, *page 89*

### Tuesday (cont.)

#### 8:30-12:00

- Tutorial 25: Introduction to Concurrent Programming in Java, *page 61*
- Tutorial 26: Reflection in Java, *page 62*
- Tutorial 29: .NET Primer with C#, *page 63*
- Workshop 29: Pair Programming Explored, *page 88*

#### 13:30-17:00

- Tutorial 33: Concurrency Utilities: Tools for Managing Multithreading, *page 65*
- Tutorial 34: C++ Idioms, *page 66*
- Tutorial 35: What’s Happening Inside Your EJB Application Server?, *page 66*

### Wednesday, 6 November

#### 8:30-10:00

- Invited Speaker: The Metaphor Metaphor, *page 28*

#### 10:00-17:30

- Tutorial 39: Ruby in a Day, *page 69*
- Tutorial 40: Aspect-Oriented Programming with AspectJ, *page 69*

#### 10:00-17:00

- CodeFest, *page 89*

#### 10:00-13:15

- Tutorial 42: Refactoring To Patterns, *page 71*
- Tutorial 43: EJB Roleplay, *page 71*

#### 10:30-12:00

- Panel: Resolved: Objects Have Failed (Part 1 of 2), *page 30*

#### 13:30-17:00

- Tutorial 46: Patterns for EJB Development, *page 72*

#### 13:30-15:00

- Technical Papers: Components and Composition, *page 31*

#### 15:30-17:00

- Technical Papers: Languages, *page 32*

## PROGRAMMING (CONT.)

### Thursday, 7 November

#### 8:30-10:00

- Turing Lecture: The Development of the Key Object-Oriented Concepts, *page 34*

#### 10:00-17:00

- CodeFest, *page 89*

#### 13:30-17:00

- Tutorial 49: How to Use Design Patterns In Java and .NET, *page 74*
- Tutorial 53: Framework Design and Implementation using Java and UML, *page 76*

#### 13:30-15:00

- Technical Papers: Aspects, *page 35*

#### 15:30-17:00

- Technical Papers: Static Analysis, *page 36*

### Friday, 8 November

#### 8:30-10:00

- Panel: Object-Oriented Success Stories, *page 38*

#### 10:00-13:00

- CodeFest, *page 89*

#### 15:30-17:00

- Technical Papers: Tools, *page 43*
- Panel: Objects and Real Time: Collision or Spiritual Union, *page 44*
- Panel: Resolved: Objects Have Failed (Part 2 of 2), *page 44*

#### 17:00-18:00

- DesignFest: DesignFest/CodeFest Wrap-Up, *page 89*

## TESTING

### Monday, 4 November

#### 8:30-17:00

- Tutorial 4: Automated Software Testing: Hands On and Interactive!, *page 48*

#### 13:30-17:00

- Workshop 31: Expanding the Boundaries of Unit Testing, *page 83*

### Wednesday, 6 November

#### 10:00-13:15

- Tutorial 44: Test-Driven Development in .NET, *page 71*

#### 13:30-17:00

- Tutorial 48: Use Cases and Testing: Using Use Cases to Write Test Cases, *page 73*

#### 15:30-17:00

- Technical Papers: Languages, *page 32*

### Thursday, 7 November

#### 13:30-17:00

- Tutorial 50: Daily Builds are for Wimps, *page 74*

#### 15:30-17:00

- Practitioner Reports: In the Trenches: Tough Problems, *page 37*



## EDUCATION

### Monday, 4 November

#### 8:30-17:00

- Workshop 8: The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- Workshop 23: Using Play to Enhance Learning About Objects, *page 82*
- Workshop 25: “Killer Examples” for Design Patterns and Objects First, *page 82*

### Tuesday, 5 November

#### 8:30-17:30

- Educators’ Symposium, *page 90*
- Doctoral Symposium, *page 92*

#### 8:30-17:00

- Workshop 21: Software Apprenticeship: the Journey to Mastery, *page 87*

### Friday, 8 November

#### 10:30-12:00

- Panel: The Failure of Object Education: What went wrong? What can we do differently?, *page 41*

## FOUNDATIONS

### Thursday, 7 November

#### 15:30-17:00

- Technical Papers: Static Analysis, *page 36*

### Friday, 8 November

#### 10:30-12:00

- Technical Papers: Foundations, *page 40*

## PATTERNS

### Monday, 4 November

#### 8:30-17:00

- Tutorial 3: Pattern-Oriented Software Architectures for Networked and Concurrent Applications, *page 48*
- Workshop 8: The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- Workshop 12: Patterns for Software Architecture, *page 81*

#### 13:30-17:00

- Tutorial 15: Patterns of Enterprise Application Architecture, *page 55*

### Tuesday, 5 November

#### 8:30-17:00

- Workshop 11: Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- Workshop 13: Patterns for Customer Interaction and Expectation Management, *page 85*

#### 8:30-12:00

- Tutorial 27: Patterns at Work, *page 62*

#### 13:30-17:00

- Tutorial 32: Dungeons and Patterns!, *page 65*
- Tutorial 36: Patterns and Application Experiences for Real-time Object Request Brokers, *page 67*

### Wednesday, 6 November

#### 10:00-13:15

- Tutorial 42: Refactoring To Patterns, *page 71*

#### 13:30-17:00

- Tutorial 45: Object-Oriented Reengineering: Patterns & Techniques, *page 72*
- Tutorial 46: Patterns for EJB Development, *page 72*

### Thursday, 7 November

#### 13:30-17:00

- Tutorial 49: How to Use Design Patterns In Java and .NET, *page 74*
- Tutorial 53: Framework Design and Implementation using Java and UML, *page 76*

#### 13:30-15:00

- Technical Papers: Aspects, *page 35*

### Friday, 8 November

#### 10:30-12:00

- Technical Papers: Scalability, *page 39*
- Technical Papers: Foundations, *page 40*

# TECHNICAL PROGRAM

**Program Chair: Satoshi Matsuoka, Tokyo Institute of Technology and National Institute of Informatics**



This year's OOPSLA presents an impressive and eclectic array of invited speakers, including the ACM Turing award lecture by Kristen Nygaard and O.J. Dahl, the pioneers of objects, and a presentation by none other than Bill Gates, the most prominent figure in the computing industry. Other speakers include OOPSLA's "ol' boy" Kent Beck; Anders Hejlsberg of Microsoft covering C#, while Alfred Z. Spector of IBM presenting the latest in Web Services; and finally, the future

of software and civilization is discussed by Jerry Michalski.

OOPSLA's technical sessions have also been renowned for not only embodying paper sessions that present the latest, top-rate academic research results, but also numerous number of panels (eight this year) that facilitate interactive dialogues by prominent leaders of the field, practitioner reports that illustrates real-world uses and solutions using object technology, posters that cover early but promising work, and demos where research prototypes as well as commercial products are demonstrated. New this year is "Onward!" that tries to identify new research directions in objects as well as software in general, and the "Web Services track" where a series of tutorials, panels, and presentations cover the latest in the next-generation software technology.

## Technical Program At A Glance

### Wednesday, 6 November

---

#### 8:30-10:00

- Invited Speaker: The Metaphor Metaphor, *page 28*

---

#### 10:30-12:00

- Technical Papers: Storage Management, *page 29*
- Panel: Resolved: Objects Have Failed (Part 1 of 2), *page 30*
- Practitioner Reports: Being Agile, *page 30*

---

#### 13:30-15:00

- Technical Papers: Components and Composition, *page 31*
- Panel: The Emperor's New Design, *page 31*
- Onward! Papers: New Models for Software I, *page 31*

---

#### 15:30-17:00

- Technical Papers: Languages, *page 32*
- Web Services Panel: If I Had a Hammer: Viewing Web Services as Your Favorite Nail, *page 32*
- Onward! Panel: Biologically Inspired Software, *page 32*

---

#### 19:30-21:00

- Onward! Invited Speaker: What's Next?, *page 33*

## Technical Program At A Glance (cont.)

### Thursday, 7 November

---

#### 8:30-10:00

- Turing Lecture: The Development of the Key Object-Oriented Concepts, *page 34*

---

#### 10:30-12:00

- Technical Papers: Optimizations, *page 34*
- Web Services Panel: Postcards from the Bleeding Edge: Web Service User/Developer Experiences, *page 35*
- Onward! Papers: New Models for Software II, *page 35*

---

#### 13:30-15:00

- Invited Speaker: The C# Programming Language, *page 35*
- Technical Papers: Aspects, *page 35*
- Onward! Papers: New Programming Constructs, *page 36*

---

#### 15:30-17:00

- Technical Papers: Static Analysis, *page 36*
- Panel: To Be Extreme, or Not to Be Extreme, *page 37*
- Practitioner Reports: In the Trenches: Tough Problems, *page 37*

### Friday, 8 November

---

#### 8:30-10:00

- Web Services Invited Speaker: Delivering on the promise of distributed systems, *page 38*
- Panel: Object-Oriented Success Stories, *page 38*

---

#### 10:30-12:00

- Technical Papers: Scalability, *page 39*
- Technical Papers: Foundations, *page 40*
- Practitioner Reports: Architecture Experiences, *page 40*

---

#### 12:00-13:00

- Onward! Panel: New Programming Constructs Beyond Inheritance, Patterns, and Notation: What's left?, *page 41*

---

#### 13:30-15:00

- Web Services Invited Speaker: The Future of Programming in a World of Web Services, *page 42*

---

#### 15:30-17:00

- Technical Papers: Tools, *page 43*
- Panel: Objects and Real Time: Collision or Spiritual Union, *page 44*
- Panel: Resolved: Objects Have Failed (Part 2 of 2), *page 44*

# INVITED SPEAKERS

**Wednesday 8:30-10:00**

**The Metaphor Metaphor, page 28**

Kent Beck, *Three Rivers Institute*

**Wednesday 19:30-21:00**

**Onward! Keynote: What's Next?, page 33**

Jerry Michalski, *Sociate*

**Thursday 8:30-10:00**

**Turing Lecture: The Development of the Key Object-Oriented Concepts, page 34**

Kristen Nygaard, *Senior Researcher, Norwegian Computing Center, Professor (emeritus) University of Oslo*  
Ole-Johan Dahl, *Professor (emeritus), University of Oslo*

**Thursday 13:30-15:00**

**The C# Programming Language, page 35**

Anders Hejlsberg, *Microsoft Corporation*

**Friday 8:30-10:00**

**Web Services: Delivering on the promise of distributed systems, page 38**

Dr. Alfred Z. Spector, *IBM Research*

**Friday 13:30-15:00**

**Web Services: The Future of Programming in a World of Web Services, page 42**

William H. Gates, *Microsoft Corporation*



# TECHNICAL PAPERS

Technical paper presentations discuss new research as well as valuable empirical results in object-oriented languages, systems, and applications. The 25 technical papers were selected after a rigorous peer review of 125 submissions by an international program committee consisting of 27 experts representing various areas of object technology. Each paper was assigned to an average of four reviewers, and papers co-authored by members of the program committee were held to higher standards, with as much as eight reviewers assigned under strict anonymity basis. The entire committee met in person for two days at the Microsoft campus in Redmond, WA, and each paper was discussed in detail, being judged on their respective novelty and their degree of contribution to object technology. The papers thus selected should advance the state-of-art of objects in significant ways.

**Wednesday 10:30-12:00**

## Storage Management, page 29

**Chair:** Tony Hosking, *Purdue University*

### *Reconsidering Custom Memory Allocation*

Emery Berger, *The University of Texas at Austin*  
Benjamin Zorn, *Microsoft Research*  
Kathryn McKinley, *The University of Texas at Austin*

### *Creating and Preserving Locality of Java Applications at Allocation and Garbage Collection Times*

Yefim Shuf, *Princeton University*  
Manish Gupta, *IBM T. J. Watson Research Center*  
Hubertus Franke, *IBM T. J. Watson Research Center*  
Andrew Appel, *Princeton University*  
Jaswinder Pal Singh, *Princeton University*

### *BuddyCache: High Performance Object Storage for Collaborative Strong-Consistency Applications in a WAN*

Magnus Bjornsson, *Brandeis University*  
Liuba Shrira, *Brandeis University*

**Wednesday 13:30-15:00**

## Components and Composition, page 31

**Chair:** Jennifer Hamilton, *Microsoft*

### *Workflow Enactment with Continuation and Future Objects*

Dragos Manolescu, *Independent consultant*

### *Integrating Independent Components with On-Demand Remodularization*

Mira Mezini, *Darmstadt Technical University*  
Klaus Ostermann, *Siemens AG*

**Wednesday 15:30-17:00**

## Languages, page 32

**Chair:** Guy Steele, *Sun Microsystems*

### *Portable Serialization of CORBA Objects: A Reflective Approach*

Marc-Olivier Killijian, *LAAS-CNRS*  
Juan-Carlos Ruiz, *LAAS-CNRS*  
Jean-Charles Fabre, *LAAS-CNRS*

### *Multiple Instances and Symbolic Variables in Executable Sequence Charts*

Rami Marelly, *Weizmann Institute of Science*  
David Harel, *Weizmann Institute of Science*  
Hillel Kugler, *Weizmann Institute of Science*

### *A Constraint-Based Architecture for Local Search*

Laurent Michel, *Brown University*  
Pascal Van Hentenryck, *Brown University*

**Thursday 10:30-12:00**

## Optimizations, page 34

**Chair:** Craig Chambers, *University of Washington*

### *Online Feedback-Directed Optimization of Java*

Matthew Arnold, *Rutgers University*  
Michael Hind, *IBM T. J. Watson Research Center*  
Barbara G. Ryder, *Rutgers University*

### *Lock Reservation: Java Locks Can Mostly Do Without Atomic Operations*

Kiyokuni Kawachiya, *IBM Tokyo Research Laboratory*  
Akira Koseki, *IBM Tokyo Research Laboratory*  
Tamiya Onodera, *IBM Tokyo Research Laboratory*

### *Fast Algorithm for Creating Space Efficient Dispatching Tables with Application to Multi-Dispatching*

Yoav Zibin, *Technion*  
Yossi Gil, *Technion*

**Thursday 13:30-15:00**

## Aspects, page 35

**Chair:** Shigeru Chiba, *Tokyo Institute of Technology*

### *Design Pattern Implementation in Java and AspectJ*

Jan Hannemann, *University of British Columbia*  
Gregor Kiczales, *University of British Columbia*

### *Implementing Distribution and Persistence Aspects with AspectJ*

Sergio Soares, *Federal University of Pernambuco*  
Eduardo Laureano, *Federal University of Pernambuco*  
Paulo Borba, *Federal University of Pernambuco*

**Thursday 15:30-17:00**

### Static Analysis, page 36

**Chair:** David Bacon, *IBM Research*

#### *Write Barrier Removal by Static Analysis*

Karen Zee, *MIT Laboratory for Computer Science*

Martin Rinard, *MIT Laboratory for Computer Science*

#### *Ownership Types for Safe Programming: Preventing Data Races and Deadlocks*

Chandrasekhar Boyapati, *MIT Laboratory for Computer Science*

Robert Lee, *MIT Laboratory for Computer Science*

Martin Rinard, *MIT Laboratory for Computer Science*

#### *An Analyzable Annotation Language*

Sarfraz Khurshid, *MIT Laboratory for Computer Science*

Darko Marinov, *MIT Laboratory for Computer Science*

Daniel Jackson, *MIT Laboratory for Computer Science*

**Friday 10:30-12:00**

### Scalability, page 39

**Chair:** Doug Lea, *SUNY Oswego*

#### *Performance and Scalability of EJB Applications*

Emmanuel Cecchet, *Rice University*

Julie Marguerite, *Rice University*

Willy Zwaenepoel, *Rice University*

#### *Incommunicado: Efficient Communication for Isolates*

Krzysztof Palacz, *Purdue University*

Greg Czajkowski, *Sun Microsystems*

Laurent Daynes, *Sun Microsystems*

Jan Vitek, *Purdue University*

#### *Sifting Out The Mud: Low Level C++ Code Reuse*

Bjorn De Sutter, *Ghent University*

Bruno De Bus, *Ghent University*

Koen De Bosschere, *Ghent University*

**Friday 10:30-12:00**

### Foundations, page 40

**Chair:** Jens Palsberg, *Purdue University*

#### *Ownership, Encapsulation, and the Disjointness of Type and Effect*

Dave Clarke, *Utrecht University*

Sophia Drossopoulou, *Imperial College London*

#### *Alias Annotations for Program Understanding*

Jonathan Aldrich, *University of Washington*

Valentin Kostadinov, *University of Washington*

Craig Chambers, *University of Washington*

#### *Towards a Formalization for COM, Part I: The Primitive Calculus*

Riccardo Pucella, *Cornell University*

**Friday 15:30-17:00**

### Tools, page 43

**Chair:** Allen Wirfs-Brock, *Instantiations*

#### *GCspy: An Adaptable Heap Visualisation Framework*

Tony Printezis, *University of Glasgow*

Richard Jones, *University of Kent*

#### *Access Rights Analysis for Java*

Larry Koved, *IBM T. J. Watson Research Center*

Marco Pistoia, *IBM T. J. Watson Research Center*

Aaron Kershenbaum, *IBM T. J. Watson Research Center*

#### *Language-Specific Make Technology for the Java Programming Language*

Mikhail Dmitriev, *Sun Microsystems Laboratories*



# PANELS

**Chair: Brian Foote, The Refactory, Inc.,**  
foote@refactory.com



OOPSLA Panels provide a forum where experts, practitioners, and an occasional novice can air their ideas, and watch them clash. Panels give OOPSLA attendees an opportunity to acquaint themselves with the latest thinking on contemporary object-oriented issues from a variety of perspectives. This year's program features a Grand Debate on the very fate of objects themselves, examinations of web service issues, a look at object-oriented success stories, and panels on

real-time programming, extreme programming, and refactoring. Take a break from the high-fiber diet being served up in the technical paper sessions, and join us for a relaxing, informative, and perhaps even entertaining respite on the red meat track.

**Wednesday 10:30-12:00**

## Resolved: Objects Have Failed (Part 1 of 2), page 30

**Moderator:** Martine Devos, *Avaya Labs*,  
mmdevos@research.avayalabs.com

### Arguing for:

Richard P. Gabriel, *Sun Microsystems*, rpg@dreamsongs.com  
Brian Foote, *The Refactory, Inc.*, foote@refactory.com

### Arguing against:

Guy L. Steele Jr., *Sun Microsystems*, Guy.Steele@sun.com  
James Noble, *Victoria University of Wellington, NZ*, kjax@mcs.vux.ac.nz

Two opposing teams will conduct a formal debate addressing the resolution "Objects Have Failed".

**Wednesday 13:30-15:00**

## The Emperor's New Design, page 31

**Moderator:** Donald B. Roberts, *The Refactory, Inc.*,  
droberts@refactory.com

William Opdyke, *North Central College*, opdyke@noctrl.edu  
Martin Fowler, *Thoughtworks, Inc.*, fowler@acm.org  
Joseph W. Yoder, *The Refactory, Inc.*, joeyoder@joeyoder.com

Should design be done up front or as the code is written? Refactoring has become a hot topic in recent years, primarily because of its association with Extreme Programming (XP). However, its roots lie in framework design, reuse, and evolution. This panel will look at refactoring over the years and address the social and technical issues associated with its introduction into an organization. It will discuss the designs that refactoring produces and whether or not they can succeed on large projects. This panel brings experts from the field of refactoring together with methodologists to debate this hot topic.

**Wednesday 15:30-17:00**

## Web Services: If I Had a Hammer: Viewing Web Services as Your Favorite Nail, page 32

Doug Lea, *Dept. of Computer Science, SUNY Oswego*,  
dl@cs.Oswego.edu

Joe Kiriya, *Caltech*

Jim Waldo, *Sun Microsystems*, Jim.Waldo@Sun.com

Sanjiva Weerawarana, *IBM Research*

Why are web services the wave of the future? What key issues do they face? Depending on whom you ask, you'll hear entirely different answers!

**Wednesday 15:30-17:00**

## Onward!: Biologically Inspired Software, page 32

Walter Fontana, *Santa Fe Institute*, walter@santafe.edu

Jeffrey O. Kephart, *IBM*, kephart@us.ibm.com, and others

**Thursday 10:30-12:00**

## Web Services: Postcards from the Bleeding Edge: Web Service User/Developer Experiences, page 35

**Moderator:** Bob Marcus, bobmarcus1@attbi.com

Dave Allison, *IBM Research*

Gerald Edgar, *Boeing Company*

Michael Rosen, *Iona*

Dan Fay, *Microsoft Corporation*

Real-world web service users and developers tell all in this panel! Find out what you need to know before you start developing or using a web service.

**Thursday 15:30-17:00**

## To Be Extreme, or Not to Be Extreme, page 37

**Moderator:** Steven Fraser, *Consultant*, sdfraser@acm.org

Kent Beck, *Three Rivers Institute*, kentbeck@csi.com

Ron Jeffries, *Object Mentor, Inc.*, ronjeffries@acm.org

Joshua Kerievsky, *Industrial Logic, Inc.*, Joshua@industriallogic.com

Charles Pool, *Iona*, Charles.Poole@iona.com

Jutta Eckstein, *Objects in Action*, jutta@acm.org

Rob Mee, *Pivotal Computer Systems*, robmee@hotmail.com

Erik Lundh, *Compelcon*, erik.lundh@compelcon.se

R. Reintz, rreintz@earthlink.net

Xtreme Programming is built on a mantra of 12 software engineering best-practices. Debate continues on the need to apply all 12 practices in order for a project to succeed.

**Friday 8:30-10:00**

### Object-Oriented Success Stories, [page 38](#)

**Moderator:** Joseph W. Yoder, *The Refactory, Inc.*,  
[yoder@refactory.com](mailto:yoder@refactory.com)  
 Ralph E. Johnson, *University of Illinois at Urbana-Champaign*,  
[Johnson@cs.uiuc.edu](mailto:Johnson@cs.uiuc.edu)

Beneath the buzz surrounding methodologies, languages and technologies, the last seventeen years have seen countless object-oriented success stories, large and small. This fishbowl will provide OOPSLA attendees with a chance to bear witness to these victories, and tell these tales at last.

**Friday 10:30-12:00**

### The Failure of Object Education: What went wrong? What can we do differently?, [page 41](#)

**Moderator:** Helen Sharp, *The Open University*, [h.c.sharp@open.ac.uk](mailto:h.c.sharp@open.ac.uk)  
**Proposer of the motion:** David West, *New Mexico Highlands University*, [dwest@cs.nmhu.edu](mailto:dwest@cs.nmhu.edu)

This is a goldfish bowl discussion around the contention that object education has failed.

**Friday 12:00-13:00**

### Onward!: New Programming Constructs Beyond Inheritance, Patterns, and Notation: What's left?, [page 41](#)

Henrik Gedenryd, *The Open University*, [h.gedenryd@open.ac.uk](mailto:h.gedenryd@open.ac.uk)  
 and others

**Friday 15:30-17:00**

### Objects and Real Time: Collision or Spiritual Union, [page 44](#)

**Moderator:** Laura Hill, *Sun Microsystems Laboratories*,  
[Laura.Hill@Sun.com](mailto:Laura.Hill@Sun.com)  
 Doug Jensen, *Mitre Corporation*, [jensen@real-time.org](mailto:jensen@real-time.org)  
 Doug Schmidt, *DARPA*, [dschmidt@darpa.mil](mailto:dschmidt@darpa.mil)  
 Raj Rajkumar, *Carnegie Mellon University*, [raj@ece.cmu.edu](mailto:raj@ece.cmu.edu)  
 Bran Selic, *Rational Software Corporation*, [bselic@rational.com](mailto:bselic@rational.com)  
 Greg Bollella, *Sun Microsystems Laboratories*

Over the past few years there has been a trend in the embedded software development community to move software development to more advanced object-oriented languages and runtimes. The movement is occasioned by a systemic increase in the complexity and size of the software required by commercially successful embedded devices. One of the distinguishing characteristics of embedded software is that it requires predictable execution of logic, i.e., it is real-time. Coincident with the above trend has been the development of real-time versions of object-oriented languages, modeling tools, and runtimes. This panel offers the current state of the development of real-time object oriented languages, modeling tools and their current and proposed uses. We discuss the Real-Time Specification for Java including it's distributed version, Real-Time Corba, Real-Time UML, and high-level real-time programming models.

**Friday 15:30-17:00**

### Resolved: Objects Have Failed (Part 2 of 2), [page 44](#)

**Moderator:** Martine Devos, *Avaya Labs*,  
[mmdevos@research.avayalabs.com](mailto:mmdevos@research.avayalabs.com)  
**Arguing for:**  
 Richard P. Gabriel, *Sun Microsystems*, [rpg@dreamsongs.com](mailto:rpg@dreamsongs.com)  
 Brian Foote, *The Refactory, Inc.*, [foote@refactory.com](mailto:foote@refactory.com)  
**Arguing against:**  
 Guy L. Steele Jr., *Sun Microsystems*, [Guy.Steele@sun.com](mailto:Guy.Steele@sun.com)  
 James Noble, *Victoria University of Wellington, NZ*, [kjx@mcs.vux.ac.nz](mailto:kjx@mcs.vux.ac.nz)

Two opposing teams will conduct a formal debate addressing the resolution "Objects Have Failed".

# PRACTITIONER REPORTS

**Chair: Rebecca J. Wirfs-Brock, Wirfs-Brock Associates, rebecca@wirfs-brock.com**



Practitioner reports are an opportunity to hear about others' experiences applying object-oriented technologies or related software development practices. Practitioner reports explore how concepts that sound good on paper (and at conferences!) work in the real world. They are a valuable means of sharing experiences, especially at the "bleeding edge" of technology. OOPSLA attendees can find out what it is like to adopt a new

language, use a new method, develop a common framework, maintain a large application, or migrate to object technology. Expectations, beliefs, and hopes can be validated, or dashed, by the experiences reported from real-world projects.

There are three Practitioner Report sessions. Each session runs 90 minutes, and has three presentations of about 25 minutes each. There is an opportunity for questions at the end of each presentation. The session on Wednesday presents experiences of teams trying to be agile in their development practices. Thursday's session focuses on tough development problems and challenges and how they were handled. Friday's session presents three architecture experiences.

**Wednesday 10:30-12:00**

## Being Agile, page 30

**Chair: John Schwartz, OHSU**

### *Agile Techniques to Avoid Firefighting at a Start Up*

Joseph Blotner, *Sabrix, Inc.*, joe.blotner@sabrix.com

This report discusses the agile software development discipline that enabled Sabrix to evolve from a chaotic environment to the point where the company was balancing both urgency and importance to drive product direction.

### *Making RUP Agile*

Michael Hirsch, *Zuehlke Engineering AG*, hm@zuehlke.com

This report goes into the details of what it takes to make RUP agile, how it was applied on two projects, and how it was configured.

### *Hitting the Target: Adding Interaction Design to Agile Software Development*

Jeff Patton, *Tomax Technologies*, jpatton@tomax.com

This report describes using interaction design in an agile development process. Recommendations are provided on how to incorporate bits of Interaction Design thinking into every day development and product planning.

**Thursday 15:30-17:00**

## In the Trenches: Tough Problems, page 37

**Chair: Jeffrey Wills, Valparaiso University**

### *A Common Multi-Platform Hardware Object Model*

Joseph Armstrong, *IBM Server Group*, jdarmstr@us.ibm.com

Astrid Kreissig, *IBM Server Group*, kloss@us.ibm.com

This report tells about the challenges of a major object-oriented design affecting many components and attempting code reuse between different projects on different hardware in different organizations and different development sites including different countries.

### *eXtreme Embedded - A report from the Front Line*

Gary Mueller, *Consultant*, garymueller@qwest.net

Janet Borzuchowski, *Storage Technology Corporation*, janetborzuchowsk@qwest.net

This report describes the challenges, obstacles, and successes encountered in applying eXtreme Programming on an embedded legacy product with a team of seasoned, veteran C programmers.

### *Transformation of an Application Data Layer*

Will Loew-Blosser, *Cargill, Inc.*, will\_loew-blosser@cargill.com

This report describes how tools and testing can be applied to make significant changes to a very large Smalltalk application.

**Friday 10:30-12:00**

## Architecture Experiences, page 40

**Chair: Lougie Anderson, Sabrix, Inc.**

### *Designing a Web Services Project for Maximum Value: The 90 Day Challenge*

Katherine Radeka, *Hewlett-Packard*, katherine\_radeka@hp.com

This report tells the story of how the 90 Day Challenge team delivered an end-to-end Web services solution to enterprise sales agents in 90 days, with immediate plans to extend the solution to other user groups.

### *Migrating Legacy Engineering Applications to Java*

Tom Dickens, *The Boeing Company*, thomas.p.dickens@boeing.com

This report presents the details of migrating a 3D-geometry surface modeling tool written in C and FORTRAN to Java.

### *Defining and Growing a Scientific Analysis Software Architecture*

William Ingram, *ExxonMobil*, william.ingram@exxonmobil.com

Rodney D. Brown, *ExxonMobil*, rodney.d.brown@exxonmobil.com

This report describes the vision, culture and environment that enabled the development of a product line software architecture, SALSA (Scientific Analysis System Layers Architecture), at ExxonMobil's Upstream Technical Computing Organization.

# ONWARD!

**Chair: Richard P. Gabriel, Sun Microsystems,**  
**rpg@dreamsongs.com**



The Onward! Track contains technical and philosophical papers describing new paradigms or metaphors in computing, new thinking about objects, new framings of computational problems or systems, and new technologies. Papers in the Onward! Track aren't aimed at advancing the state of the art - they're aimed, instead, at altering or redefining the art by proposing a leap forward - or sideways - for computing.

**Wednesday 13:30-15:00**

## Papers: New Models for Software I, page 31

### *A Vision of Autonomic Computing*

Jeffrey O. Kephart, IBM, [kephart@us.ibm.com](mailto:kephart@us.ibm.com)

### *Magic*

Dave West, New Mexico Highlands University, [dwest@cs.nmhu.edu](mailto:dwest@cs.nmhu.edu)

**Wednesday 15:30-17:00**

## Panel: Biologically Inspired Software, page 32

Walter Fontana, Santa Fe Institute, [walter@santafe.edu](mailto:walter@santafe.edu)

Jeffrey O. Kephart, IBM, [kephart@us.ibm.com](mailto:kephart@us.ibm.com),  
 and others

**Wednesday 19:30-21:00**

## Onward! Keynote: What's Next?, page 33

Jerry Michalski, Sociate

**Thursday 10:30-12:00**

## Papers: New Models for Software II, page 35

### *Notes on Postmodern Programming*

James Noble, Victoria University of Wellington, New Zealand,  
[kjx@mcs.vuw.ac.nz](mailto:kjx@mcs.vuw.ac.nz)

Robert Biddle, Victoria University of Wellington, New Zealand,  
[robert@mcs.vuw.ac.nz](mailto:robert@mcs.vuw.ac.nz)

### *Principles of Lean Thinking*

Mary Poppendieck, Poppendieck.LLC, [mary@poppendieck.com](mailto:mary@poppendieck.com)

**Thursday 13:30-15:00**

## Papers: New Programming Constructs, page 36

### *Problematic Encapsulation in High-Risk Systems*

Daniel Dvorak, Jet Propulsion Laboratory, California Institute of  
 Technology, [daniel.dvorak@jpl.nasa.gov](mailto:daniel.dvorak@jpl.nasa.gov)

### *Many-to-Many Invocation*

Alan Kaminsky, Rochester Institute of Technology, [ark@it.rit.edu](mailto:ark@it.rit.edu)

Hans-Peter Bischof, Rochester Institute of Technology, [hpb@cs.rit.edu](mailto:hpb@cs.rit.edu)

**Friday 12:00-13:00**

## Panel: New Programming Constructs Beyond Inheritance, Patterns, and Notation: What's left?, page 41

Henrik Gedenryd, The Open University, [h.gedenryd@open.ac.uk](mailto:h.gedenryd@open.ac.uk),  
 and others

# OBJECTS AND WEB SERVICES

**Chairs: Brent Hailpern and Peri Tarr,**  
**IBM T. J. Watson Research Center,**  
**oows2002@watson.ibm.com**



Way back in the 20th century, everyone got really excited by a “new” concept called “object oriented.” It was a new way to think about building systems, a new way to describe components, a new way to achieve reuse, a new way to build models and write

programs. Turned out OO caused almost as many problems as it solved! The next “new” concept / structure / description / reuse / model / language is coming (or it is here) and it is called “Web Services.” The Objects and Web Services Technical Track at OOPSLA 2002 will be a week-long event to put Web Services under the microscope, to see where it overlaps with our OO experiences, and to see if we can, as a community, build a common future for it and objects. The week will encompass eight tutorials, two workshops, two panels, technical presentations, and two plenary talks all devoted to the relationships between the exciting new area of web services and the core focus of OOPSLA, object-oriented systems.

Web Services are pieces of software that provide access to their capabilities across the World Wide Web via XML interfaces. The use of XML enables systems running in different environments and in different locations to exchange information, interoperate, and be combined more readily than ever before. Web Services hold out the dream of a truly composable distributed system where “applications” are built out of “components” provided by other companies or individuals and where these applications are services that can adapt and change over time to meet the dynamic needs of web users.

Web services are, therefore, all about promoting sharing of functionality and information among heterogeneous systems, and about addressing the question, how do they work together? Objects, on the other hand, are all about promoting the development and evolution of high-quality software, through mechanisms such as encapsulation, non-invasive adaptation, reuse, patterns, and low-impact evolution-properties that will be needed more than ever in web services. How, then, do we get the best software engineering properties of objects into XML-based web services? This technical track will explore those issues and will bring together researchers and practitioners from both disciplines to hash out the problems, issues, and opportunities and object-oriented web services become a reality.

For listing of all events related to Objects and Web Services (including workshops and tutorials), please refer to the Web Services Track on *page 11*.

**Wednesday 15:30-17:00**

## **Panel: If I Had a Hammer: Viewing Web Services as Your Favorite Nail, *page 32***

Doug Lea, *Dept. of Computer Science, SUNY Oswego,*  
*dl@cs.Oswego.edu*

Joe Kiniry, *Caltech*

Jim Waldo, *Sun Microsystems, Jim.Waldo@Sun.com*

Sanjiva Weerawarana, *IBM Research*

Why are web services the wave of the future? What key issues do they face? Depending on whom you ask, you'll hear entirely different answers!

**Thursday 10:30-12:00**

## **Panel: Postcards from the Bleeding Edge: Web Service User/Developer Experiences, *page 35***

**Moderator:** Bob Marcus, *bobmarcus1@attbi.com*

Dave Allison, *IBM Research*

Gerald Edgar, *Boeing Company*

Michael Rosen, *Iona*

Dan Fay, *Microsoft Corporation*

Real-world web service users and developers tell all in this panel! Find out what you need to know before you start developing or using a web service.

**Friday 8:30-10:00**

## **Invited Speaker: Delivering on the promise of distributed systems, *page 38***

Dr. Alfred Z. Spector, *IBM Research*

**Friday 10:30-11:00**

## **Paper: Performance and Scalability of EJB Applications, *page 39***

Emmanuel Cecchet, *Rice University*

Julie Marguerite, *Rice University*

Willy Zwaenepoel, *Rice University*

**Friday 10:30-11:00**

## **Practitioner Report: Designing a Web Services Project for Maximum Value: The 90 Day Challenge, *page 40***

Katherine Radeka, *Hewlett-Packard, katherine\_radeka@hp.com*

This report tells the story of how the 90 Day Challenge team delivered an end-to-end Web services solution to enterprise sales agents in 90 days, with immediate plans to extend the solution to other user groups.

**Friday 13:30-15:00**

## **Invited Speaker: The Future of Programming in a World of Web Services, *page 42***

William H. Gates, *Microsoft Corporation*



## Details of the Technical Program

Wednesday, 6 November

8:30-10:00

### Welcome and Introduction

**Conference Chair:** Mamdouh Ibrahim, *IBM Global Services*

**Program Chair:** Satoshi Matsuoka, *Tokyo Institute of Technology* and *National Institute of Informatics*

### Invited Speaker: The Metaphor Metaphor



Kent Beck, *Three Rivers Institute*

Of all the aspects of Extreme Programming, the suggestion that customers and developers share a common metaphor or metaphors for the system is the most problematic. “We’re doing all of XP (except metaphor, of course),” read a recent email. Having tried to explain it every way I could, folks just aren’t getting it.

There are two possibilities: I have done a poor job of explaining metaphors and their importance, or I’m flat wrong and metaphors just aren’t that important. So, here’s the deal—I will try just once more to explain the positive role of consciously metaphorical thinking and the danger of letting your metaphors ride you instead of the other way around. If it works, if people get it, great. If not, I will swear off trying to explain metaphors forever.

Examples will be taken from programming, business, and (in keeping with the self-referential title of the talk) OOPSLA itself. At the end of the talk, we will take a vote on whether metaphors and their importance to programming are beginning to make sense, or if I should just shut up about it.

### Speaker

*Kent Beck is best defined in terms of relationships. Following work by Jim Coplien and Ward Cunningham on software development process, with Ron Jeffries and the C3 team at Chrysler he invented and named Extreme Programming, resulting in the Jolt Productivity Award-winning “Extreme Programming Explained: Embrace Change.” He is the co-author of “Planning Extreme Programming” with Martin Fowler, with whom he also collaborated on “Refactoring: Improving the Design of Existing Systems.” With Erich Gamma he is writing JUnit, the award-winning de facto standard testing tool for Java. With Ward Cunningham he wrote HotDraw, a widely copied drawing editor framework, pioneered patterns for software development, and popularized CRC cards. He channeled the Ancient Smalltalk Masters to produce “The Smalltalk Best Practice Patterns,” and is currently reviving a decades-old technique in the forthcoming “Test-Driven Development By Example.” He lives on a southern Oregon farm with a dwindling but still-impressive gaggle of children, his lovely wife Cindee, four dogs, and a bunch of chickens.*

10:30-12:00

**SESSION A****Technical Papers: Storage Management****Chair:** Tony Hosking, *Purdue University*

Storage Management is an inherent and important underlying, enabling technology for modern object-oriented languages and systems. The first paper proposes a new storage allocator called reaps that combines the benefits of both region-based allocation and individual, object-based deletion. The second paper aims to improve garbage collection time by exploiting the locality of objects and their allocations during traversal. The third paper considers transactional caching in a distributed objects setting and proposes a technique to exploit the cached objects in peers, thus eliminating expensive roundtrip access to remote servers.

***Reconsidering Custom Memory Allocation***Emery Berger, *The University of Texas at Austin*Benjamin Zorn, *Microsoft Research*Kathryn McKinley, *The University of Texas at Austin*

Programmers hoping to achieve performance improvements often use custom memory allocators. This in-depth study examines eight applications that use custom allocators. Surprisingly, for six of these applications, a state-of-the-art general-purpose allocator (the Lea allocator) performs as well as or better than the custom allocators. The two exceptions use regions, which deliver higher performance (improvements of up to 44%). Regions also reduce programmer burden and eliminate a source of memory leaks. However, we show that the inability of programmers to free individual objects within regions can lead to a substantial increase in memory consumption. Worse, this limitation precludes the use of regions in common programming idioms, reducing their usefulness.

We present a generalization of general-purpose and region-based allocators that we call reaps. Reaps are a combination of regions and heaps, providing a full range of region semantics with the addition of individual object deletion. We show that our implementation of reaps provides high performance, outperforming other allocators with region-like semantics. Our results indicate that programmers needing fast regions should use reaps, and that most programmers considering custom allocators should instead use the Lea allocator.

***Creating and Preserving Locality of Java Applications at Allocation and Garbage Collection Times***Yefim Shuf, *Princeton University*Manish Gupta, *IBM T. J. Watson Research Center*Hubertus Franke, *IBM T. J. Watson Research Center*Andrew Appel, *Princeton University*Jaswinder Pal Singh, *Princeton University*

The growing gap between processor and memory speeds is motivating the need for optimization strategies that improve data locality. A major challenge is to devise techniques suitable for pointer-intensive applications. This paper presents two techniques aimed at improving the memory behavior of pointer-intensive applications with dynamic memory allocation, such as those written in Java. First, we present an allocation time object placement technique based on the recently introduced notion of “prolific” (frequently instantiated) types. We attempt to co-locate, at allocation time, objects of prolific types that are connected via object references. Then, we present a novel technique for traversing live objects at garbage collection (GC) time. The benefits of this techniques are twofold: (i) it improves performance of GC due to better locality during a heap traversal and (ii) it restructures surviving objects in a way that enhances locality. On multiprocessors, this technique can further reduce overhead due to synchronization and false sharing. The experimental results, on a well-known suite of Java benchmarks (SPECjvm98, SPECjbb2000, and jOlden), from an implementation of these techniques in the Jikes RVM, are very encouraging. The object co-allocation technique improves application performance by up to 21% (10% on average) in the Jikes RVM configured with a non-copying mark-and-sweep collector. The locality-based traversal technique reduces GC times by up to 20% (10% on average) and improves performance of applications by up to 14% (6% on average) in the Jikes RVM configured with a copying semi-space collector. Both techniques combined can improve application performance by up to 22% (10% on average) in the Jikes RVM configured with a non-copying mark-and-sweep collector.

***BuddyCache: High Performance Object Storage for Collaborative Strong-Consistency Applications in a WAN***Magnus Bjornsson, *Brandeis University*Liuba Shrira, *Brandeis University*

Collaborative applications provide a shared work environment for groups of networked clients collaborating on a common task. They require strong consistency for shared persistent data and efficient access to fine-grained objects. These properties are difficult to provide in wide-area networks because of high network latency. BuddyCache is a new transactional caching approach that improves the latency of access to shared persistent objects for collaborative strong-consistency applications in high-latency network environments. The challenge is to improve performance while providing the correctness and availability properties of a transactional caching protocol in the presence of node failures and slow peers. We have implemented a BuddyCache prototype and evaluated its performance. Analytical results, confirmed by measurements of the BuddyCache prototype using the multi-user 007 benchmark indicate that for typical Internet latencies, e.g. ranging from 40 to 80 milliseconds round trip time to the storage server, peers using BuddyCache can reduce by up to 50% the latency of access to shared objects compared to accessing the remote servers directly.



## SESSION B

### Panel: Resolved: Objects Have Failed (Part 1 of 2)

**Moderator:** Martine Devos, *Avaya Labs*,  
mmdevos@research.avayalabs.com

**Arguing for:**

Richard P. Gabriel, *Sun Microsystems*, rpg@dreamsongs.com

Brian Foote, *The Refactory, Inc.*, foote@refactory.com

**Arguing against:**

Guy L. Steele Jr., *Sun Microsystems*, Guy.Steele@sun.com

James Noble, *Victoria University of Wellington, NZ*, kjax@mcs.vux.ac.nz

The Grand Debate will address the resolution, “Objects Have Failed.” The evidence is substantial. The .com meltdown was caused in no small part by the impossibility of designing and coding flexible businesses, even when substantial investments were made. Object technology has proven particularly brittle because it foregrounds structure over form, a problem inherent in many extant programming languages. Though one could argue that modern computing technology is the root cause, objects contributed more than their fair share to the debacle. Yet, a persuasive case can be made that the ascendancy of objects is, by now, a fact; a matter beyond dispute. Objects have infused every area of software technology, from graphical user interface construction to programming language design. It can be argued that object-oriented programming has become a redundant term: “Programming” will now suffice.

This debate will be presented in two parts, so as to allow the opposing teams to properly and thoroughly address the issues raised by their opponents.

## SESSION C

### Practitioner Reports: Being Agile

**Chair:** John Schwartz, *OHSU*

#### *Agile Techniques to Avoid Firefighting at a Start Up*

Joseph Blotner, *Sabrix, Inc.*, joe.blotner@sabrix.com

This report describes the creation and evolution of the (modified) agile methodology implemented at Sabrix, Inc., a start-up enterprise-class software company. The company was delivering alpha quality software to early adopter customers on a weekly basis, using urgency as the primary driver for product development. New features and bug fixes were completed in a manner similar to a soccer game between two teams of five-year-olds, with everyone on the team chasing down the latest emergency. This report discusses the agile software development discipline that enabled Sabrix to evolve from that chaotic environment to the point where the company was balancing both urgency and importance to drive product direction. This discipline enabled the company to actually deliver a “Release”, and move forward into a new product architecture.

#### *Making RUP Agile*

Michael Hirsch, *Zuehlke Engineering AG*, hm@zuehlke.com

The Unified Development Process (USDP) and especially its implementation from Rational Software Corporation, the Rational Unified Process (RUP), is a comprehensive process covering almost all aspects of software development projects. However, due to the great level of detail provided by RUP many professionals consider RUP as not practical for small and fast paced projects.

This presentation reports the experiences made with RUP in two small projects with teams of three to four developers. RUP proved to be adaptable to the needs of small projects and was very effective in both projects. One key to the successful application of RUP in small projects is the careful selection of a proper subset of artifacts and keeping these artifacts very concise and free from unnecessary formalisms. This report goes into the details of what it takes to make RUP agile, how it was applied on the two projects, and how it was configured. Also covered is what elements of RUP contributed to the success of one project, and why RUP could not prevent less than optimal results in the other project.

#### *Hitting the Target: Adding Interaction Design to Agile Software Development*

Jeff Patton, *Tomax Technologies*, jpatton@tomax.com

Extreme Programming appears to be a solution for discovering and meeting requirements faster (through close customer collaboration) as well as creating quality software. In practice we found XP did deliver high quality software quickly, but the resulting product still failed to delight the customer. Although the finished product should have been an exact fit, the actual end-user still ended up slogging through the system to accomplish necessary day-to-day work. This report describes using interaction design in an agile development process to resolve this issue. Using interaction design as a day-to-day practice throughout an iterative development process helps our team at Tomax Technologies deliver high quality software, while feeling confident the resulting software will more likely meet end-user expectations. The method of Interaction Design followed here is based on Constantine and Lockwood’s Usage-Centered Design. Recommendations are provided on how to practice an agile form of U-CD and how to incorporate bits of Interaction Design thinking into every day development and product planning decisions.

13:30-15:00

**SESSION A****Technical Papers: Components and Composition****Chair:** Jennifer Hamilton, *Microsoft*

Components and program construction via their compositions are becoming accepted practices of software development. However, component composition often does not work straightforwardly when mismatches occur, resulting in burdensome efforts by the programmers. The first paper focuses on reusable workflow components, and introduces various concepts from programming language theory to resolve the impedance mismatch. The second paper facilitates the separation of an application into independent reusable building blocks, and deals with the integration of pre-build generic software components that have been developed by third party vendors via their dynamic, on-demand remodularizations.

***Workflow Enactment with Continuation and Future Objects***Dragos Manolescu, *Independent consultant*

An increasing number of software developers are turning to workflow to separate the logic and the control aspects in their applications, thus making them more amenable to change. However, in spite of recent efforts to standardize and provide reusable workflow components, many developers build their own. This is a challenging endeavor and involves solving problems which seem incompatible with the object paradigm and current object-oriented programming languages. In the context of an object-oriented workflow framework, this paper demonstrates a novel approach that resolves this impedance mismatch with techniques drawn from programming language theory. This successful cross-pollination narrows the gap between the results of decades of research in programming languages and developers working hard to cope with change.

***Integrating Independent Components with On-Demand Remodularization***Mira Mezini, *Darmstadt Technical University*Klaus Ostermann, *Siemens AG*

This paper proposes language concepts that facilitate the separation of an application into independent reusable building blocks and the integration of pre-build generic software components into applications that have been developed by third party vendors. A key element of our approach are on-demand remodularizations, meaning that the abstractions and vocabulary of an existing code base are translated into the vocabulary understood by a set of components that are connected by a common collaboration interface. This general concept allows us to mix-and-match remodularizations and components on demand.

**SESSION B****Panel: The Emperor's New Design****Moderator:** Donald B. Roberts, *The Refactory, Inc.*,*droberts@refactory.com*William Opdyke, *North Central College*, *opdyke@noctrl.edu*Martin Fowler, *fowler@acm.org*Joseph W. Yoder, *The Refactory, Inc.*, *joeyoder@joeyoder.com*

Should design be done up front or as the code is written? Refactoring has become a hot topic in recent years, primarily because of its association with Extreme Programming (XP). However, its roots lie in framework design and reuse. This panel will look at refactoring over the years and address the social and technical issues associated with its introduction into an organization. They will discuss the designs that refactoring produces and whether or not they can succeed on large projects. This panel brings experts from the field of refactoring together with methodologists to debate this hot topic.

**SESSION C****Onward!: New Models for Software I**

This session considers new ways of looking at software. The first paper describes the vision behind work being done at IBM to make computing more resilient, and the second invites us to look at computing through the metaphorical lens of magic.

***A Vision of Autonomic Computing***Jeffrey O. Kephart, *IBM*, *kephart@us.ibm.com****Magic***Dave West, *New Mexico Highlands University*, *dwest@cs.nmhu.edu*

15:30-17:00

**SESSION A****Technical Papers: Languages****Chair:** Guy Steele, *Sun Microsystems*

This session is a collection of innovative language-level research and techniques that advances the state-of-art in objects. The first paper aims to define, implement and illustrate a portable serialization technique for CORBA objects based on reflection, in particular using open compilers. The second paper extends the well-known design language, live sequence charts, into a powerful executable language thereby allowing detailed description of object interaction in a succinct fashion. The third paper combines the traditional constraint-solving language and object-oriented language to derive a powerful framework for solving combinatorial optimization problems.

***Portable Serialization of CORBA Objects: A Reflective Approach***Marc-Olivier Killijian, *LAAS-CNRS*Juan-Carlos Ruiz, *LAAS-CNRS*Jean-Charles Fabre, *LAAS-CNRS*

The objective of this work is to define, implement and illustrate a portable serialization technique for CORBA objects based on a reflective approach. Through open compilers facilities the internal state of CORBA objects is obtained and transformed into a portable format thanks to CORBA facilities. This state can be restored and used by objects developed using different languages and running on different software platforms. A tool was developed and applied to a Chat application as a case study. The proposed technique is used to exchange state information between a C++ and a Java incarnation of this CORBA service. An observer tool enables the object state to be displayed and analyzed by the user. The applicability of this technique to various domains is also discussed. We finally advocate that reflection (and related techniques) is a powerful concept to extend the work presented in this paper.

***Multiple Instances and Symbolic Variables in Executable Sequence Charts***Rami Marelly, *Weizmann Institute of Science*David Harel, *Weizmann Institute of Science*Hillel Kugler, *Weizmann Institute of Science*

We extend live sequence charts (LSCs), a highly expressive variant of sequence diagrams, and provide the extension with an executable semantics. The extension involves support for instances that can bind to multiple objects and symbolic variables that can bind to arbitrary values. The result is a powerful executable language for expressing behavioral requirements on the level of inter-object interaction. The extension is implemented in full in our play-engine tool, with which one can execute the requirements directly without the need to build or synthesize an intra-object system model. It seems that in addition to many advantages in testing and requirements engineering, for some kinds of systems this could lead to the requirements actually serving as the final implementation.

***A Constraint-Based Architecture for Local Search***Laurent Michel, *Brown University*Pascal Van Hentenryck, *Brown University*

Combinatorial optimization problems are ubiquitous in many practical applications. Yet most of them are challenging, both from computational complexity and programming standpoints. Local search is one of the main approaches to address these problems. However, it often requires sophisticated incremental algorithms and data structures, and considerable experimentation. This paper proposes a constraint-based, object-oriented, architecture to reduce the development time of local search algorithms significantly. The architecture consists of declarative and search components. The declarative component includes invariants, which maintain complex expressions incrementally, and differentiable objects, which maintain properties that can be queried to evaluate the effect of local moves. Differentiable objects are high-level modeling concepts, such as constraints and functions, that capture combinatorial substructures arising in many applications. The search component supports various abstractions to specify heuristics and meta-heuristics. We illustrate the architecture with the language Comet and several applications, such as car sequencing and the progressive party problem. The applications indicate that the architecture allows for very high-level modeling of local search algorithms, while preserving excellent performance.

**SESSION B****Web Services Panel: If I Had a Hammer: Viewing Web Services as Your Favorite Nail**Doug Lea, *Dept. of Computer Science, SUNY Oswego*,  
*dl@cs.Oswego.edu*Joe Kiniry, *Caltech*Jim Waldo, *Sun Microsystems*, *Jim.Waldo@Sun.com*Sanjiva Weerawarana, *IBM Research*

Why are web services the wave of the future? What key issues do they face? Depending on whom you ask, you'll hear entirely different answers! This panel will give you 5 different viewpoints on the topic of web services, from the perspective of distributed systems, programming languages, software engineering, patterns, security, tools, business, and standards. At present, we envision posing one or more problems in creating, integrating, or evolving web services to the panelists, which each panelist will solve from a different perspective, each highlighting different critical issues. The panelists will discuss the differences and perhaps debate the relative merits of the different solutions. Along the way, we expect to begin to answer the question, "Why will web services succeed when all previous 'standard' distributed services and languages have failed to thrive (such as DCE, CORBA, DCOM, etc.)?"

**SESSION C****Onward! Panel: Biologically Inspired Software**Walter Fontana, *Santa Fe Institute*, *walter@santafe.edu*Jeffrey O. Kephart, *IBM*, *kephart@us.ibm.com*, and others

This panel considers using biological metaphors and processes for computing and organizing computations.

19:30-21:00

**Invited Speaker Onward! Keynote: What's Next?**

Jerry Michalski, *Sociate*

Technology is not neutral. It reflects the objectives and mental models of those who design it, the business imperatives of the times and the interactions of those who use it. By tracing the history behind some of today's critical technologies, then describing the dynamics between the major forces in the business market and the market of ideas, Jerry will tackle questions such as:

- Why does it seem that innovation is at a standstill, despite much emphasis on corporate innovation?
- What role do our assumptions about capitalism, intellectual property, assets and scarcity have in our continuing evolution?
- How will programming quickly/slowly, in the large/in the small, with closed/open models and with highly structured/unstructured, organic approaches play out?
- Where should technology developers place their energies?

**Speaker**

*Jerry Michalski, formerly managing editor of the monthly technology newsletter Release 1.0, currently a freelance consultant and soon to be a Harvard Business School Press author, will present an unconventional look at what's next for programming - and civilization.*





## Thursday, 7 November

8:30-10:00

### Turing Lecture: The Development of the Key Object-Oriented Concepts



Kristen Nygaard, *Senior Researcher, Norwegian Computing Center, Professor (emeritus) University of Oslo*  
Ole-Johan Dahl, *Professor (emeritus), University of Oslo*

The lecture will trace in detail the development of the key object-oriented concepts: objects, classes, quasi-parallel (really multi-threaded) program execution, inheritance (classes and subclasses) and virtual entities, and describe the setting in which this happened."

#### Speakers

*Kristen Nygaard is professor (emeritus) of informatics at the University of Oslo, and senior researcher at the Norwegian Computing Center. Together with Ole-Johan Dahl he developed the key object-oriented programming concepts at the Norwegian Computing Center from 1961-67. Professor Nygaard was involved in large-scale simulation studies at the Norwegian Defense Research Establishment from 1949-60. He continued his work on object-orientation, and did research on systems development (the "Scandinavian School"), participative system design, and societal consequences of information technology. With Danish colleagues, he invented Beta, a general object-oriented language. He is now active in the COOL Project (Comprehensive Object-Oriented Learning), developing new tools for teaching object-orientation.*

*Professor Dahl also worked at the Norwegian Defense Research Establishment, and joined the Simula project as an experienced designer and implementer of basic software as well as high level programming language. In 1968, Dahl became the first professor of informatics at the University of Oslo, responsible for establishing research and education programs in this rapidly expanding field. After Simula, his focus on computer program verification led to the development of his theory of constructive types and subtypes based on computer-aided concept formation and reasoning.*

*The lecture will be presented by Professor Kristen Nygaard.*

10:30-12:00

## SESSION A

### Technical Papers: Optimizations

**Chair:** Craig Chambers, *University of Washington*

This session showcases a collection of new optimization techniques to significantly improve the runtime performance of object-oriented programs. The first paper proposes a generalized feedback-directed optimization system that requires no prior instrumentation, describing ways how traditional as well as new optimizations are driven by feedback information, and measures their effectiveness. The second paper proposes a novel algorithm for reducing thread locking overhead called "lock reservation", by which Java thread locks are reserved in advance before they are actually used, exploiting locality of lock usage by threads. The third paper shows an extension of an efficient single-dispatch algorithm into a multi-dispatch variant, and demonstrates its time and space efficiency.

#### Online Feedback-Directed Optimization of Java

Matthew Arnold, *Rutgers University*

Michael Hind, *IBM T. J. Watson Research Center*

Barbara G. Ryder, *Rutgers University*

This paper describes the implementation of a general online feedback-directed optimization system. The system is fully automatic; it requires no prior profiling run. It uses a previously developed low-overhead instrumentation sampling framework to collect control flow graph edge profiles. This profile information is used to drive several traditional optimizations, as well as a novel algorithm for performing feedback-directed splitting. We empirically evaluate this system and demonstrate improvements in peak performance of up to 20% while overhead remains low, with no individual execution being degraded by more than 2% because of initial instrumentation.

#### Lock Reservation: Java Locks Can Mostly Do Without Atomic Operations

Kiyokuni Kawachiya, *IBM Tokyo Research Laboratory*

Akira Koseki, *IBM Tokyo Research Laboratory*

Tamiya Onodera, *IBM Tokyo Research Laboratory*

Because of the built-in support for multi-threaded programming, Java programs perform many lock operations. Although the overhead has significantly been reduced in the recent virtual machines, one or more atomic operations are required for locking and unlocking an object even in the fastest cases. This paper presents a novel algorithm called "lock reservation". It exploits "thread locality" of Java locks, which claims that the locking sequence of a Java lock contains a very long sequence of a specific thread. The algorithm allows locks to be reserved for threads. When a thread attempts to acquire a lock, it can do without any atomic operation if the lock is reserved for the thread. Otherwise, it cancels the reservation and falls back to a conventional locking algorithm. We have evaluated an implementation of lock reservation in IBM's production virtual machine and compiler. The results show that it achieved performance improvements up to 28% in real Java programs.

### ***Fast Algorithm for Creating Space Efficient Dispatching Tables with Application to Multi-Dispatching***

Yoav Zibin, *Technion*  
Yossi Gil, *Technion*

The dispatching problem can be solved very efficiently in the single-inheritance setting. In this paper we show how to extend one such solution to the multiple-inheritance setting. This generalization comes with an increase to the space requirements by a small factor of  $n$ Slices. This factor can be thought of as a metric of the complexity of the topology of the inheritance hierarchy. On a data set of 35 hierarchies totaling some 64 thousand types, our dispatching data structure, based on a novel type slicing technique, exhibits very significant improvements over previous dispatching techniques, not only in terms of the time for creating the underlying data structure, but also in terms of total space used. The cost is in the dispatching time, which is no longer constant, but doubly logarithmic in the number of types. Conversely, by using a simple binary search, dispatching time is logarithmic in the number of different implementations. In practice dispatching uses one indirect branch and, on average, only 2.5 binary branches. Our results also have applications to the space-efficient implementation of the more general problem of dispatching multi-methods.

## **SESSION B**

### **Web Services Panel: Postcards from the Bleeding Edge: Web Service User/Developer Experiences**

**Moderator:** Bob Marcus, *bobmarcus1@attbi.com*  
Dave Allison, *IBM Research*  
Gerald Edgar, *Boeing Company*  
Michael Rosen, *Iona*  
Dan Fay, *Microsoft Corporation*

Real-world web service users and developers tell all in this panel! Find out what you need to know before you start developing or using a web service. Some questions that will be answered include: What is the current status of attempts to develop and deploy real live Web Services? What parts of the technology are mature? What are the open issues and future requirements? What role does object technology play in production implementations? What works? What doesn't? Why did the panelists choose it? Would they do it again?

## **SESSION C**

### **Onward!: New Models for Software II**

This is a session on new approaches to conceptualizing programming. The first paper is a postmodern work examining ideas of postmodernism in the computing realm, and the second is a look at agile methodologies as manifestations of Lean Thinking.

#### ***Notes on Postmodern Programming***

James Noble, *Victoria University of Wellington, New Zealand*,  
*kjx@mcs.vuw.ac.nz*  
Robert Biddle, *Victoria University of Wellington, New Zealand*,  
*robert@mcs.vuw.ac.nz*

#### ***Principles of Lean Thinking***

Mary Poppendieck, *Poppendieck.LLC*, *mary@poppendieck.com*

13:30-15:00

## **SESSION A**

### **Invited Speaker: The C# Programming Language**



Anders Hejlsberg, *Microsoft Corporation*  
C# (pronounced "C Sharp") is a new programming language introduced with the Microsoft .NET platform and recently standardized by ECMA. This talk gives an overview of the C# language and examines some of its key design concepts, such as component-oriented programming support, unified type system, extensible metadata, and versioning.

#### **Speaker**

*Anders Hejlsberg is a Distinguished Engineer in the Developer Division at Microsoft Corporation. He is the chief designer of the C# programming language and a key participant in the development of the .NET Framework. Before joining Microsoft in 1996, Anders was a Principal Engineer at Borland International. As one of the first employees of Borland, he was the original author of Turbo Pascal and later worked as the Chief Architect of the Delphi product line. Anders studied Engineering at the Technical University of Denmark.*

## **SESSION B**

### **Technical Papers: Aspects**

**Chair:** Shigeru Chiba, *Tokyo Institute of Technology*

Aspect-oriented programming (AOP) is a novel paradigm in software engineering and programming languages. It provides programmers with a powerful technique for modularizing crosscutting concerns, which cut across multiple classes or components. In this session, two papers reporting on programming experiences with AspectJ, which is a seamless aspect-oriented extension to Java and is one of the most popular AOP languages. The first paper shows that AspectJ implementations of Gang-of-Four (GoF) design patterns improve textual locality, reusability, and composability in comparison to pure Java implementations. The second paper illustrates their experiences of implementation of object distribution and persistence in AspectJ. It also proposes minor extensions to AspectJ that they discovered to be potentially useful during their development.

#### ***Design Pattern Implementation in Java and AspectJ***

Jan Hannemann, *University of British Columbia*  
Gregor Kiczales, *University of British Columbia*

Implementing the GOF design patterns in both Java and AspectJ shows that in many cases the implementations can be improved with aspect-oriented techniques. The improvements fall into three main categories: textual locality, reusability, and composability. The degree of improvement in implementation varies, with the greatest improvement coming when the pattern solution structure involves crosscutting of some form, including one object playing multiple roles, many objects playing one role, or an object playing roles in multiple pattern instances.

### ***Implementing Distribution and Persistence Aspects with AspectJ***

Sergio Soares, *Federal University of Pernambuco*  
 Eduardo Laureano, *Federal University of Pernambuco*  
 Paulo Borba, *Federal University of Pernambuco*

This paper presents guidelines for implementing or refactoring distribution and persistence code using AspectJ, a general-purpose aspect-oriented extension to Java. These guidelines were derived from our experience on implementing distribution and persistence aspects in a real information system, a health complaint system, which was originally implemented in Java and restructured with AspectJ. We also identify a few drawbacks in the language and suggest some minor modifications that could significantly improve the implementations that could be derived by following our guidelines. Some of the aspects implemented in our experiment constitute a simple AspectJ framework, whereas others are application specific. Nevertheless we show that different implementations might follow a common AspectJ pattern. The framework and the pattern allow us to provide the general guidelines for both restructuring and implementing persistent and distributed applications with AspectJ.

## **SESSION C**

### **Onward!: New Programming Constructs**

This session considers new ways of looking at software. The first paper by a member of the Mars Lander team is a critical look at the handcuffs encapsulation can put on design, and the second presents a new way to look at Web Services.

#### ***Problematic Encapsulation in High-Risk Systems***

Daniel Dvorak, *Jet Propulsion Laboratory, California Institute of Technology*, [daniel.dvorak@jpl.nasa.gov](mailto:daniel.dvorak@jpl.nasa.gov)

#### ***Many-to-Many Invocation***

Alan Kaminsky, *Rochester Institute of Technology*, [ark@it.rit.edu](mailto:ark@it.rit.edu)  
 Hans-Peter Bischof, *Rochester Institute of Technology*, [hpb@cs.rit.edu](mailto:hpb@cs.rit.edu)

15:30-17:00

## **SESSION A**

### **Technical Papers: Static Analysis**

**Chair:** David Bacon, *IBM Research*

Static program analysis is the basis for most of the powerful optimization techniques behind modern high-performance programming languages. In this session, static analysis is applied to solve a wide range of problems, from traditional compiler optimization to automated verification of program properties. The first paper shows that it is usually possible to eliminate almost all dynamically executed write barriers from a program by using static information about allocation order combined with code reordering techniques. The second paper extends the authors' previous work on race-free Java to the problem of statically preventing deadlocks. Finally, the third paper shows how high-level program properties can be specified as annotations to Java programs and then checked automatically.

#### ***Write Barrier Removal by Static Analysis***

Karen Zee, *MIT Laboratory for Computer Science*  
 Martin Rinard, *MIT Laboratory for Computer Science*

We present a new static analysis for removing unnecessary write barriers in programs that use generational garbage collection. To our knowledge, this is the first analysis for this purpose. Our algorithm uses an interprocedural, flow-sensitive pointer analysis to locate assignments that always create a reference from a younger object to an older object, then transforms the program to eliminate the write barriers normally associated with such assignments. To enhance the effectiveness of the technique, we have implemented two transformations that reorder object allocations. These transformations can dramatically increase the number of write barriers that our algorithm is able to eliminate. Results from our implemented system show that our technique can eliminate the majority of the write barriers in most of the programs in our benchmark set, producing modest performance improvements of up to 8% of the overall execution time. Moreover, by dynamically instrumenting the executable, we are able to show that for all but two of our nine benchmark programs, our analysis is close to optimal in the sense that it eliminates the write barriers for almost all assignments that do not, in the observed execution, create a reference from an older object to a younger object.

#### ***Ownership Types for Safe Programming: Preventing Data Races and Deadlocks***

Chandrasekhar Boyapati, *MIT Laboratory for Computer Science*  
 Robert Lee, *MIT Laboratory for Computer Science*  
 Martin Rinard, *MIT Laboratory for Computer Science*

This paper presents a new static type system for multithreaded programs; well-typed programs in our system are guaranteed to be free of data races and deadlocks. Our type system allows programmers to partition the locks into a fixed number of equivalence classes and specify a partial order among the equivalence classes. The type checker then statically verifies that whenever a thread holds more than one lock, the thread acquires the locks in the descending order. Our system also allows programmers to use recursive tree-based data structures to describe the partial order. For example, programmers can specify that nodes in a tree must be locked in the tree-order. Our system allows mutations in the data structure that change the partial order at runtime. The type checker statically verifies that the mutations do not introduce cycles in the partial order, and that the changing of the partial order does not lead to deadlocks. We do not know of any other sound static system for preventing deadlocks that allows changes to the partial order at runtime.



**An Analyzable Annotation Language**

Sarfraz Khurshid, *MIT Laboratory for Computer Science*  
 Darko Marinov, *MIT Laboratory for Computer Science*  
 Daniel Jackson, *MIT Laboratory for Computer Science*

The Alloy Annotation Language (AAL) is a language for annotating Java code based on the Alloy modeling language. It offers a syntax similar to the Java Modeling Language (JML), and the same opportunities for generation of run-time assertions. In addition, however, AAL offers the possibility of fully automatic compile-time analysis. Several kinds of analyses are supported, including: checking the code of a method against its specification; checking that the specification of a method in a subclass is compatible with the specification in the superclass; and checking properties relating method calls on different objects, such as that the equals methods of a class (and its overrides) induce an equivalence. Using partial models in place of code, it is also possible to analyze object-oriented designs in the abstract: investigating, for example, a view relationship amongst objects. The paper gives examples of annotations and such analyses. It presents (informally) a systematic translation of annotations into Alloy, a simple first-order logic with relational operators. By doing so, it makes Alloy's automatic analysis, which is based on state-of-the-art SAT solvers, applicable to the analysis of object-oriented programs, and demonstrates the power of a simple logic as the basis for an annotation language.

**SESSION B****Panel: To Be Extreme, or Not to Be Extreme**

**Moderator:** Steven Fraser, *Consultant*, [sdfraser@acm.org](mailto:sdfraser@acm.org)  
 Kent Beck, *Three Rivers Institute*, [kentbeck@csi.com](mailto:kentbeck@csi.com)  
 Ron Jeffries, *ronjeffries@acm.org*  
 Joshua Kerievsky, *Industrial Logic, Inc.*, [Joshua@industriallogic.com](mailto:Joshua@industriallogic.com)  
 Charles Pool, *Iona*, [Charles.Poole@iona.com](mailto:Charles.Poole@iona.com)  
 Jutta Eckstein, [jutta@acm.org](mailto:jutta@acm.org)  
 Rob Mee, [robmee@hotmail.com](mailto:robmee@hotmail.com)  
 Erik Lundh, [erik.lundh@compelcon.se](mailto:erik.lundh@compelcon.se)  
 R. Reintz, [rreintz@earthlink.net](mailto:rreintz@earthlink.net)

eXtreme Programming is built atop a canon of 12 software engineering best-practices. Debate continues on the necessity of application of all 12 practices for the success of a project. Heresy (religious methodological differences) continues in the community on the subject of "missing practices", i.e. teams that have reported success while not embracing all XP practices.

This panel will provide a lively (you WILL have fun), informative, highly interactive session to debate differences and similarities in approach. Audience members will be questioned at regular intervals for their opinion.

We propose panelists from a variety of contexts (corporate, consulting, academic) to share their opinions, biases, and experiences.

**SESSION C****Practitioner Reports: In the Trenches: Tough Problems**

**Chair:** Jeffrey Wills, *Valparaiso University*

**A Common Multi-Platform Hardware Object Model**

Joseph Armstrong, *IBM Server Group*, [jdarmstr@us.ibm.com](mailto:jdarmstr@us.ibm.com)  
 Astrid Kreissig, *IBM Server Group*, [kloss@us.ibm.com](mailto:kloss@us.ibm.com)

About 5 years ago one group in IBM's high end server system started a redesign of its hardware access layer. Flexibility for any kind of configuration and hardware was the main goal for the design, to allow for rapid bringup changes and changing hardware packaging.

Object-oriented design was the obvious choice. About 2 years ago another group in IBM's middle and low end server systems started a redesign of their hardware access layer. The same basic objectives as the high end systems applied. Additionally the middle and low end systems group was continuing the consolidation of its servers to one eServer with a common service processor. Both hardware platforms are multi-processor systems supporting tera-bytes of I/O and a multitude of operating systems from z/OS and OS/400 to AIX and Linux.

Everything cried for reuse.

This report is about the challenges of a major object-oriented design affecting many components and attempting code reuse between different projects on different hardware in different organizations and different development sites including different countries. We will list the goals we had, where we are now, what proved to be good concepts and what the roadblocks were. The report ends with an outlook on what we will approach next.

**eXtreme Embedded - A report from the Front Line**

Gary Mueller, *Consultant*, [garymueller@qwest.net](mailto:garymueller@qwest.net)  
 Janet Borzuchowski, *Storage Technology Corporation*,  
[janetborzuchowsk@qwest.net](mailto:janetborzuchowsk@qwest.net)

Many embedded development environments are stuck somewhere in the backwash of software technology. Structured programming practices, with the artifacts of those methodologies liberally scattered about, are the norm in these environments. The world has changed since structured methodologies first emerged, with shorter and shorter market windows and increasingly sophisticated customers demanding more and more capabilities at lower and lower cost. Embedded system development must also adapt to these market imperatives to stay competitive. This report describes the challenges, obstacles, and successes encountered in applying eXtreme Programming on an embedded legacy product with a team of seasoned, veteran C programmers.

**Transformation of an Application Data Layer**

Will Loew-Blosser, *Cargill, Inc.*, [will\\_loew-blosser@cargill.com](mailto:will_loew-blosser@cargill.com)

Changing a fundamental interface in a large application is typically considered impractical because of high risks and costs. This report demonstrates that with careful use of tools and testing, risks and costs can be significantly reduced. A very large Smalltalk application was developed at Cargill. About 2,000 classes in this application interacted with a data access framework. The framework dynamically performed runtime mapping of object attributes to data table columns. Analysis showed that although dynamic look up consumed 40% of execution time, it was unnecessary.

To improve performance, we decided to replace dynamic lookup with an explicit object to database mapping scheme. Our solution involved changing the data framework as well as the application classes. The challenge was how to change 2000 classes in a systematic manner.

Normally, refactoring consists of a series of small safe steps that improve design while preserving behavior. Instead of performing many small transformations, we used the facilities in Brant and Roberts' Smalltalk Refactoring browser to make more significant program transformations. Using the facilities in Smalltalk Refactoring Browser we defined our own program transformation rules and then applied them.

This allowed us to systematically make 16,000 changes almost bug free. We further reduced our risk by rigorously testing our changes using Beck's S-Unit tool. After these transformations, execution time spent in the data access framework was reduced from 40% to less than 2%.

## Friday, 8 November

8:30-10:00

### SESSION A

#### Web Services Invited Speaker: Delivering on the promise of distributed systems



Dr. Alfred Z. Spector, *IBM Research*

The study of distributed systems in computer science is now many decades old. While concepts of directories, remote procedure call, distributed load balancing, etc. have been around a long time, it has been very difficult to create a successful general purpose platform that supports real distributed applications. In various generations of technology, we have seen the continual reinvention of RPC (e.g., DCE,

DCOM, CORBA, RMI, and IIOP) not to mention all the variations on message passing, transaction processing, security models, etc. that seem necessary for real distributed applications. In each generation, we have improved on the flexibility and dynamic nature of distributed interactions, with an eye to increasing their generality and commercial viability. Yet each generation has also stumbled on the way to achieving a usable, open, heterogeneous, distributed environment. Some systems failed because of incompleteness; some because of complexity; some because of inadequate tools; and others because they were proprietary or could not support heterogeneity.

Finally, with the advent of Web Services from the W3C, the promise of a widely accepted, open, heterogeneous, distributed environment may be at hand. Significantly, Web Services depend on the key proposition taught to us by the object oriented community - that scalability and designability depend on encapsulation (with clean interfaces) to reduce complexity. But we must also learn from the dark side of OO: we already know that good tools, useability, and performance will make or break the success of Web Services. Experiences from the history of both distributed systems and object orientation can teach us quite a lot about how to succeed this time with Web Services. This talk will place Web Services in the context of distributed computing's history and propose a roadmap for how it can succeed as the basis for the open platform for distributed computing in the future.

#### Speaker

*Dr. Alfred Z. Spector is vice president of Services and Software in IBM's Research Division responsible for setting IBM's worldwide research strategy in support of its Services and Software businesses. Recently, Dr. Spector was an Adjunct Professor at Columbia University's Computer Science Department and Senior Technical Strategist in IBM's Application and Integration Middleware (AIM) business, which has responsibility for a number of IBM software product families including CICS, WebSphere, MQSeries, WebSphere Studio and Eclipse. Previously, Dr. Spector was the general manager of Marketing and Strategy for IBM's AIM business, and the general manager of IBM's Transaction Systems business. Dr. Spector was also founder and CEO of Transarc Corporation, a pioneer in distributed transaction processing (Encina, DCE) and wide area file systems (DFS, AFS), and an Associate Professor of Computer Science at Carnegie Mellon University. Prior to joining Carnegie Mellon, Dr. Spector was affiliated with the IBM San Jose (now Almaden) Research Laboratory while in graduate school. Dr. Spector received his Ph.D. in Computer Science from Stanford University on the topic of multiprocessing architectures for local area computer networks and his A.B. in Applied Mathematics, Magna cum Laude, from Harvard University.*

### SESSION B

#### Panel: Object-Oriented Success Stories

**Moderator:** Joseph W. Yoder, *The Refactory, Inc.*,

*yoder@refactory.com*

Ralph E. Johnson, *University of Illinois at Urbana-Champaign*,  
*Johnson@cs.uiuc.edu*

Beneath the buzz around methodologies, languages and technologies, the last seventeen years have seen countless object-oriented success stories, large and small. This fishbowl will provide OOPSLA attendees to bear witness to these victories, and tell these tales at last.

10:30-12:00

**SESSION A****Technical Papers: Scalability****Chair:** Doug Lea, *SUNY Oswego*

This session covers three very different aspects of software scalability. The first paper measures the performance consequences of using different styles of enterprise Java applications and servers. The second describes a set of mechanisms for reducing communication overhead for Java remote invocations among otherwise isolated tasks running on the same computer system. The third paper takes scalability in the opposite direction. It describes link-time analyses and techniques that can reduce code sizes of C++ applications so that they can be deployed on small devices.

***Performance and Scalability of EJB Applications***Emmanuel Cecchet, *Rice University*Julie Marguerite, *Rice University*Willy Zwaenepoel, *Rice University*

We investigate the combined effect of application implementation method, container design, and efficiency of communication layers on the performance scalability of J2EE application servers by detailed measurement and profiling of an auction site server.

We have implemented three versions of the auction site. The first version uses stateless session beans with bean-managed persistence, making only minimal use of the services provided by the EJB container. The second version makes extensive use of the container services using entity beans with container-managed persistence. The third version applies the session facade pattern, using session beans as a facade to access entity beans. We evaluate these different implementations on two popular open-source EJB containers with orthogonal designs: JBoss uses dynamic proxies to generate the container classes at run time, making an extensive use of reflection. JOnAS pre-compiles classes during deployment, minimizing the use of reflection at run time. We also evaluate the communication optimizations provided by each of these EJB containers.

The most important factor in determining performance is the implementation method. EJB applications with session beans perform as well as a Java servlet implementation and an order-of-magnitude better than most of the implementations based on entity beans. Use of session facade beans improves performance for entity beans, but only if local communication is very efficient. Otherwise, session facade beans degrade performance.

For the implementation using session beans, communication cost forms the major component of the execution time on the EJB server. The design of the container has little effect on performance. For implementations using session facade beans, local communication cost is critically important. With entity beans, the design of the container becomes important as well. In particular, the cost of reflection affects performance.

***Incommunicado: Efficient Communication for Isolates***Krzysztof Palacz, *Purdue University*Greg Czajkowski, *Sun Microsystems*Laurent Daynes, *Sun Microsystems*Jan Vitek, *Purdue University*

Colocating computations in a single instance of safe-language virtual machine can improve performance and overall platform scalability. It also poses various challenges. One of them is providing a very fast inter-application communication mechanism. In addition to being efficient, such a mechanism should not violate any functional and non-functional properties of its environment, and should also support enforcement of application-specific security policies. This paper explores the design and implementation of a communication substrate for applications executing within a single Java virtual machine. In particular, our goal is to provide a minimal software layer for fast and secure communication between isolated computations in the context of the Multitasking Virtual Machine. Designing an efficient extension that does not break isolation properties and at the same time pragmatically offers an API very closely resembling this of the Java Remote Method Invocation has proven non-trivial. This paper demonstrates a set of techniques that lead to at least an eight-fold performance improvement over the in-process inter-application communication using standard mechanisms offered by the Java platform.

***Sifting Out The Mud: Low Level C++ Code Reuse***Bjorn De Sutter, *Ghent University*Bruno De Bus, *Ghent University*Koen De Bosschere, *Ghent University*

More and more computers are being incorporated in devices where the available amount of memory is limited. This contrasts with the increasing need for additional functionality and the need for rapid application development. While object-oriented programming languages, providing mechanisms such as inheritance and templates, allow fast development of complex applications, they have a detrimental effect on program size. This paper introduces new techniques to reuse the code of whole procedures at the binary level and a supporting technique for data reuse. These techniques benefit specifically from program properties originating from the use of templates and inheritance. Together with our previous work on code abstraction at lower levels of granularity, they achieve additional code size reductions of up to 38% on already highly optimized and compacted binaries, without sacrificing execution speed. We have incorporated these techniques in Squeeze++, a prototype link-time binary rewriter for the Alpha architecture, and extensively evaluate them on a suite of 8 real-life C++ applications. The total code size reductions achieved post link-time (i.e. without requiring any change to the compiler) range from 27 to 70%, averaging at around 43%.

## SESSION B

### Technical Papers: Foundations

**Chair:** Jens Palsberg, *Purdue University*

The papers in this session present mathematical foundations for reasoning about object-oriented software. Each paper gives a type-based analysis that can prove key properties of a program. The first paper concerns proving that two statements have disjoint effects and that two expressions cannot lead to aliases. The second paper presents an extension of Java in which one can specify and check information about owners and ownership of objects. The third paper shows an approach to modeling components and enforcing COM conformance. Each paper contains key correctness theorems, including type soundness.

#### ***Ownership, Encapsulation, and the Disjointness of Type and Effect***

Dave Clarke, *Utrecht University*

Sophia Drossopoulou, *Imperial College London*

Ownership types provide a statically enforceable notion of object-level encapsulation in object-oriented programs. We extend ownership types with computational effects, so as to support reasoning about object-oriented programs. The ensuing system itself provides both access control and effects reporting. Based on this type system, we codify two formal systems for reasoning about aliasing and the disjointness of computational effects. The first can be used to prove that evaluation of two expressions will never lead to aliases, while the latter can be used to show the noninterference of two expressions.

#### ***Alias Annotations for Program Understanding***

Jonathan Aldrich, *University of Washington*

Valentin Kostadinov, *University of Washington*

Craig Chambers, *University of Washington*

Building and evolving a large software system is challenging in part because it is difficult to understand the system architecture: how the system breaks down into parts, and how those parts interact. One of the primary challenges in understanding the software architecture of an object-oriented program is dealing with aliasing between objects. Unexpected aliasing can lead to broken invariants, mistaken assumptions, security holes, and surprising side effects, all of which may lead to software defects and complicate software evolution. This paper presents AliasJava, an alias annotation system for Java that makes alias patterns explicit in the source code, enabling developers to reason more effectively about the interactions in a complex system. We describe our implementation, prove the soundness of the annotation system, and give an algorithm for automatically inferring alias annotations. Our experience suggests that the annotation system is practical, that annotation inference yields appropriate annotations, and that the annotations can express important invariants of data structures and of software architectures.

#### ***Towards a Formalization for COM, Part I: The Primitive Calculus***

Riccardo Pucella, *Cornell University*

We introduce in this paper a typed calculus intended to capture the execution model of COM. The innovation of this calculus is to model very low-level aspects of the COM framework, specifically the notion of interface pointers. This is handled by specifying an allocation semantics for the calculus, thereby modeling heap allocation of interfaces explicitly. Having an explicit way of talking about interface pointers allows us to model in a reasonable way the notions of interface sharing and object identity. We introduce a type system that can be used to disambiguate between specification and implementation of interfaces. The type system moreover can capture a notion of COM conformance, that is, the legality of COM components. We discuss extensions of the calculus to handle subtyping of interfaces, dynamic interface negotiation and aggregation.

## SESSION C

### Practitioner Reports: Architecture Experiences

**Chair:** Lougie Anderson, *Sabrix, Inc.*

#### ***Designing a Web Services Project for Maximum Value: The 90 Day Challenge***

Katherine Radeka, *Hewlett-Packard*, [katherine\\_radeka@hp.com](mailto:katherine_radeka@hp.com)

The 90 Day Challenge team set out to deliver an end-to-end Web services solution to enterprise sales agents in 90 days, with immediate plans to extend the solution to other user groups. In order to meet their commitments, the team would have to reuse the same component Web services in multiple contexts. By modeling the workflows and carefully managing scope at the component level, the team developed a suite of Web services that other teams could easily incorporate into multiple projects. As a result, three other programs to date have incorporated Web services from the 90 Day Challenge, saving weeks of development time.

#### ***Migrating Legacy Engineering Applications to Java***

Tom Dickens, *The Boeing Company*, [thomas.p.dickens@boeing.com](mailto:thomas.p.dickens@boeing.com)

The Boeing Company, like many other engineering-centric companies, has a large base of legacy applications written in FORTRAN and C. In today's computing environment, maintaining and evolving these applications is becoming difficult. One such Boeing application, the Aero Grid and Paneling System (AGPS), is a 3D-geometry surface modeling tool. In the fall of 2001 we completed the migration of the AGPS source code from 300,000 lines of mixed C and FORTRAN to 150,000 lines of 100% Java. The migration resulted in many benefits; some anticipated and some unexpected. The benefits include widespread portability of AGPS on engineering workstations and PC-class machines, allowing AGPS to embrace many modern programming capabilities readily available in Java, greatly improving the maintainability and enhanceability of the AGPS source code, and an increased robustness of the AGPS code. During the migration we also took the opportunity to rearchitect major areas of the code using object-oriented techniques, to modernize the graphical user interface, and to implement a significant number of pending enhancements from our backlog.

This report looks into the details of the AGPS migration to Java, discussing our observations, lessons learned, migration techniques, and what we see in the future computing environment. Data are presented and discussed to substantiate our conclusions that Java is well-suited for compute-intensive engineering applications, Java portability is a reality, Java performance is no longer a problem, and that embracing object-oriented programming techniques produces a much superior product with less effort.



## Defining and Growing a Scientific Analysis Software Architecture

William Ingram, *ExxonMobil*, [william.ingram@exxonmobil.com](mailto:william.ingram@exxonmobil.com)  
 Rodney D. Brown, *ExxonMobil*, [rodney.d.brown@exxonmobil.com](mailto:rodney.d.brown@exxonmobil.com)

The computing employed in oil and gas exploration is predominately scientific, resulting in a variety of data analysis applications. Although the analytical domains vary greatly (e.g., seismic processing, geologic modeling, engineering facilities design, etc.), the requirements that shape their software architectures are similar. Such analysis systems are rarely illustrated in the software analysis/design and architecture literature.

We describe a product line software architecture, SALSA (Scientific Analysis System Layers Architecture), developed at ExxonMobil's Upstream Technical Computing Organization. Key forces that led to the SALSA include the data analysis process itself, the need to access data stored in a variety of formats, integrating our analysis tools with third party tools, and isolating our codes from external changes. Although derived from Layers software architectures described in the literature, we have elaborated SALSA to identify objects within each layer specific to an architecture focused on analysis

SALSA is a product of a software development cultural change instituted over the past few years at ExxonMobil. We adopted a vision of "architecture-centric" software development. This vision, supported at the upper management and grassroots levels, began with a software architecture team to steward our architectures; and a career development program including OOAD training for all developers and a biweekly "School of the Architects" seminar series. Thus far, the results include a way of thinking about our main product, captured in SALSA; several product-line architectures based on SALSA; a beginning of an architecture-centric reuse effort, also based on SALSA; and a visible change in the way we view and approach software development.

## SESSION D

### Panel: The Failure of Object Education: What went wrong? What can we do differently?

**Moderator:** Helen Sharp, *The Open University*, [h.c.sharp@open.ac.uk](mailto:h.c.sharp@open.ac.uk)

**Proposer of the motion:** David West, *New Mexico Highlands University*, [dwest@cs.nmhu.edu](mailto:dwest@cs.nmhu.edu)

Efforts to teach objects have failed. Evidence to support this claim includes the following. There are few, if any, examples of software that we can say were truly developed using the object-oriented approach. Few people, practitioners or academics, really understand (or agree upon) what an object is. The diluted and co-opted form of "objects" that comprise mainstream development are but pale imitations of the radical innovation implicit in the original conceptualization of objects.

This session is a continuation of a discussion started at the Educators' Symposium on Tuesday, and is included in the main program to open the debate to a wider audience. Participants will discuss these issues, hearing opinions for and against these claims, suggestions for what (if anything) has gone wrong with object education, and what (if anything) we can do about it.

12:00-13:00

## SESSION A

### Onward! Panel: New Programming Constructs Beyond Inheritance, Patterns, and Notation: What's left?

Henrik Gedenryd, *The Open University*, [h.gedenryd@open.ac.uk](mailto:h.gedenryd@open.ac.uk)  
*and others*

This panel looks at what we can hope for in new programming constructs or new programming languages.

13:30-15:00

**Web Services Invited Speaker: The Future of Programming in a World of Web Services**

William H. Gates, *Microsoft Corporation*

While the discipline of programming has evolved over more than 50 years of modern computing, the human programmer still bears most of the burden of designing, coding, and testing a system or application under development. Moreover, the programmer is generally tasked with making some assessment or guarantee of a system's quality, security and reliability. With the emergence of

Web services and new types of development systems, what will the experience of building applications be in 5 years? 10 years? How do Web services move us closer to a world of systems controlling and even programming other systems? What will the programming interfaces of the future--both the languages and the environments--look like?

In his keynote address, Microsoft Chief Software Architect Bill Gates will address these questions and more. The keynote will also feature demonstrations of innovative concepts, tools and technologies being developed by scientists and engineers from Microsoft Research and Microsoft product development groups.

**Speaker**

*William (Bill) H. Gates is chairman and chief software architect of Microsoft Corporation, the worldwide leader in software, services and Internet technologies for personal and business computing. Microsoft had revenues of \$25.3 billion for the fiscal year ending June 2001, and employs more than 40,000 people in 60 countries. In his junior year, Gates left Harvard University to devote his energies to Microsoft, a company he had begun in 1975 with his childhood friend Paul Allen. Guided by a belief that the computer would be a valuable tool on every office desktop and in every home, they began developing software for personal computers. Gates' foresight and his vision for personal computing have been central to the success of Microsoft and the software industry. Under Gates' leadership, Microsoft's mission has been to continually advance and improve software technology, and to make it easier, more cost-effective and more enjoyable for people to use computers. The company is committed to a long-term view, reflected in its investment of more than \$4 billion on research and development in the current fiscal year.*

15:30-17:00

**SESSION A****Technical Papers: Tools****Chair:** Allen Wirfs-Brock, *Instantiations*

Software Tools are now important part of application development, and they involve sophisticated software technologies that rival those of the languages and systems themselves. The first paper presents a framework for visualizing and observing the dynamic behavior of the storage heap, adaptable to a variety of garbage-collected as well as non-garbage collected languages. The second paper presents a technique for computing the access rights requirements by using a context sensitive, flow sensitive, interprocedural data flow analysis for the Java 2 security architecture. The third paper describes the language-specific “make” technology for Java, and demonstrates that the algorithm used therein is superior to conventional dependency tracking tools employed in Unix systems.

***GCspy: An Adaptable Heap Visualisation Framework***Tony Printezis, *University of Glasgow*Richard Jones, *University of Kent*

GCspy is an architectural framework for the collection, transmission, storage and replay of memory management behaviour. It makes new contributions to the understanding of the dynamic memory behaviour of programming languages (and especially object-oriented languages that make heavy demands on the performance of memory managers). GCspy’s architecture allows easy incorporation into any memory management system: it is not limited to garbage-collected languages. It requires only small changes to the system in which it is incorporated but provides a simple to use yet powerful data-gathering API. GCspy scales to allow very large heaps to be visualised effectively and efficiently. It allows already-running, local or remote, systems to be visualised and those systems to run at full speed outside the points at which data is gathered. GCspy’s visualisation tool presents this information in a number of novel ways.

Deep understanding of program behaviour is essential to the design of the next generation of garbage collectors and explicit allocators. Until now no satisfactory tools have been available to assist the implementer in gaining an understanding of heap behaviour. GCspy has been demonstrated to be a practical solution to this dilemma. It has been used to analyse production Java virtual machines running applications of realistic size. Its use has revealed important insights into the interaction between application program and JVM and has led to the development of better garbage collectors.

***Access Rights Analysis for Java***Larry Koved, *IBM T. J. Watson Research Center*Marco Pistoia, *IBM T. J. Watson Research Center*Aaron Kershenbaum, *IBM T. J. Watson Research Center*

Java 2 has a security architecture that protects systems from unauthorized access by mobile or statically configured code. The problem is in manually determining the set of security access rights required to execute a library or application. The commonly used strategy is to execute the code, note authorization failures, allocate additional access rights, and test again. This process iterates until the code successfully runs for the test cases in hand. Test cases usually do not cover all paths through the code, so failures can occur in deployed systems. Conversely, a broad set of access rights is allocated to the code to prevent authorization failures from occurring. However, this often leads to a violation of the “principle of least privilege.” This paper presents a technique for computing the access rights requirements by using a context sensitive, flow sensitive, interprocedural data flow analysis. By using this analysis, we compute at each program point the set of access rights required by the code. We model features such as multi-threading, implicitly defined security policies, the semantics of the `Permission` class, method and generation of a security policy description. We implemented the algorithms and present the results of our analysis on a set of programs. While the analysis techniques described in this paper are in the context of Java code, the basic techniques are applicable to access rights analysis issues in non-Java-based systems.

***Language-Specific Make Technology for the Java Programming Language***Mikhail Dmitriev, *Sun Microsystems Laboratories*

Keeping consistent the code of a Java application (code is consistent if all of the project classes can be recompiled together without errors) prevents late linking errors, and thus may significantly improve development turnaround time. In this paper we describe a language-specific make (smart recompilation) technology for Java, that guarantees consistency of the project code, and at the same time reduces the number of source code recompilations to the minimum. After project code consistency is initially assured by complete recompilation, the information extracted from the binary classes is stored in the so-called project database. Whenever the source code for some class C is changed, its recompiled binary is compared to the old version of C preserved in the project database. As a result, we find a minimum subset of classes that depend on C and may be affected by the particular change made to it. They are recompiled in turn, and absence of compilation errors at this phase guarantees the consistency of the new project code. To determine which dependent classes to recompile, we categorize all source incompatible changes, and for each category establish a criterion for finding the smallest possible subset of dependent classes.

## SESSION B

### Panel: Objects and Real Time: Collision or Spiritual Union

**Moderator:** Laura Hill, *Sun Microsystems Laboratories*,  
*Laura.Hill@Sun.com*

Doug Jensen, *Mitre Corporation*, *jensen@real-time.org*

Doug Schmidt, *DARPA*, *dschmidt@darpa.mil*

Raj Rajkumar, *raj@ece.cmu.edu*

Bran Selic, *bselic@rational.com*

Greg Bollella, *Sun Microsystems Laboratories*

Over the past few years there has been a trend in the embedded software development community to move software development to more advanced object-oriented languages and runtimes. The movement is occasioned by a systemic increase in the complexity and size of the software required by commercially successful embedded devices. One of the distinguishing characteristics of embedded software is that it requires predictable execution of logic, i.e., it is real-time. Coincident with the above trend has been the development of real-time versions of object-oriented languages, modeling tools, and runtimes. This panel offers the current state of the development of real-time object oriented languages, modeling tools and their current and proposed uses. We discuss the Real-Time Specification for Java including it's distributed version, Real-Time Corba, Real-Time UML, and high-level real-time programming models.

## SESSION C

### Panel: Resolved: Objects Have Failed (Part 2 of 2)

**Moderator:** Martine Devos, *Avaya Labs*,  
*mmdevos@research.avayalabs.com*

**Arguing for:**

Richard P. Gabriel, *Sun Microsystems*, *rpg@dreamsongs.com*

Brian Foote, *The Refactory, Inc.*, *foote@refactory.com*

**Arguing against:**

Guy L. Steele Jr., *Sun Microsystems*, *Guy.Steele@sun.com*

James Noble, *Victoria University of Wellington, NZ*, *kjx@mcs.vux.ac.nz*

The Grand Debate will address the resolution, "Objects Have Failed." The evidence is substantial. The .com meltdown was caused in no small part by the impossibility of designing and coding flexible businesses, even when substantial investments were made. Object technology has proven particularly brittle because it foregrounds structure over form, a problem inherent in many extant programming languages. Though one could argue that modern computing technology is the root cause, objects contributed more than their fair share to the debacle. Yet, a persuasive case can be made that the ascendancy of objects is, by now, a fact; a matter beyond dispute. Objects have infused every area of software technology, from graphical user interface construction to programming language design. It can be argued that object-oriented programming has become a redundant term: "Programming" will now suffice.

This debate will be presented in two parts, so as to allow the opposing teams to properly and thoroughly address the issues raised by their opponents.



# TUTORIALS

**Chair: Sherman R. Alpert, IBM T.J. Watson Research Center, [tutorials@oopsla.acm.org](mailto:tutorials@oopsla.acm.org)**



OOPSLA 2002 continues the Conference's tradition of offering world class tutorials presented by renowned researchers and practitioners. Tutorials are presented during the first two days of the Conference as well as during the technical sessions on days three and four.

Tutorials will be presented in the Washington State Convention & Trade Center. All tutorials include a complimentary lunch and breaks with OOPSLA-supplied snacks and drinks. All tutorial

attendees are also invited to the Tutorials and Workshops Reception on Monday evening.

## Monday, 4 November

### 8:30-17:00

- 1 Essential Object-oriented Analysis and Design, *page 47*
- 2 Agile Software Development Methodologies: Principles and Practices, *page 47*
- 3 Pattern-Oriented Software Architectures for Networked and Concurrent Applications, *page 48*
- 4 Automated Software Testing: Hands On and Interactive!, *page 48*
- 5 Pragmatic Programming - the First Day, *page 49*
- 6 Programming Web Services Applications, *page 50*

### 8:30-12:00

- 7 Building Secure OO Systems - A Practical Guide and Overview, *page 51*
- 8 Introduction to the .NET Framework, *page 51*
- 9 The Art of Writing Use Cases, *page 52*
- 10 Programmer's Dozen: Thirteen Practices for Refactoring, Repairing, and Regaining Control of Your Code, *page 53*
- 11 Efficient Implementation of Object-Oriented Programming Languages, *page 53*

### 13:30-17:00

- 12 Card-based User and Use Case Task Modeling, *page 54*
- 13 Developing Java Applications for Mobile Devices, *page 54*
- 14 Object vs. The Web, *page 55*
- 15 Patterns of Enterprise Application Architecture, *page 55*
- 16 Aspect-Oriented Programming in C++, *page 56*
- 17 The Design and Implementation of the Jikes RVM Optimizing Compiler, *page 56*

## Tuesday, 5 November

### 8:30-17:00

- 18 A Brief Tour of Responsibility-Driven Design, *page 57*
- 19 Concepts of Object-Oriented Programming, *page 58*
- 20 Web Services and Service Oriented Architectures, *page 58*
- 21 A Language Designer's View of Rotor, Microsoft's "Shared Source" Implementation of the Common Language Infrastructure, *page 59*

### 8:30-12:00

- 22 Extreme Programming, A Simulation, *page 60*
- 23 Designing an Agile Methodology, *page 60*
- 24 J2EE for Enterprise Application Integration and e-business Integration, *page 61*
- 25 Introduction to Concurrent Programming in Java, *page 61*
- 26 Reflection in Java, *page 62*
- 27 Patterns at Work, *page 62*
- 28 Use Cases, Usability Requirements and User Interfaces, *page 63*
- 29 .NET Primer with C#, *page 63*

### 13:30-17:00

- 30 eXtreme Programming & Web Development, *page 64*
- 31 Patterns for Writing Effective Use Cases, *page 64*
- 32 Dungeons and Patterns!, *page 65*
- 33 Concurrency Utilities: Tools for Managing Multithreading, *page 65*
- 34 C++ Idioms, *page 66*
- 35 What's Happening Inside Your EJB Application Server?, *page 66*
- 36 Patterns and Application Experiences for Real-time Object Request Brokers, *page 67*
- 37 Objects, XML and Databases, *page 67*
- 38 Distributed .NET, *page 68*

## Wednesday, 6 November

---

### 10:00-17:30

- 39 Ruby in a Day, *page 69*
  - 40 Aspect-Oriented Programming with AspectJ, *page 69*
  - 41 Programming Web Services for Mobile Devices, *page 70*
- 

### 10:00-13:15

- 42 Refactoring To Patterns, *page 71*
  - 43 EJB Roleplay, *page 71*
  - 44 Test-Driven Development in .NET, *page 71*
- 

### 13:30-17:00

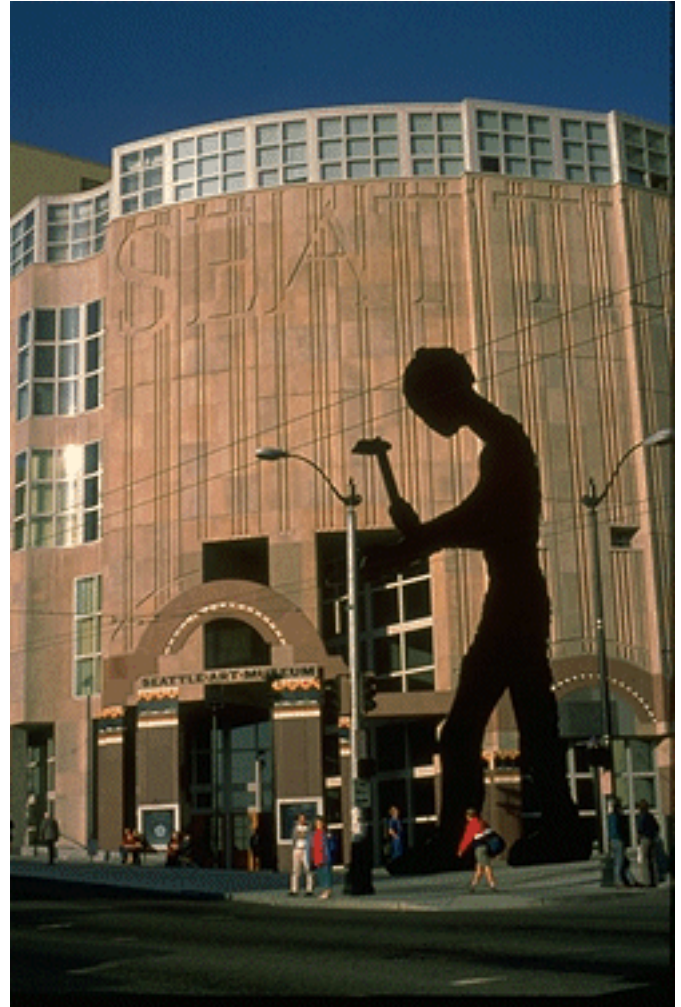
- 45 Object-Oriented Reengineering: Patterns & Techniques, *page 72*
- 46 Patterns for EJB Development, *page 72*
- 47 Scrum and Agile Process 101, *page 73*
- 48 Use Cases and Testing: Using Use Cases to Write Test Cases, *page 73*

## Thursday, 7 November

---

### 13:30-17:00

- 49 How to Use Design Patterns In Java and .NET, *page 74*
- 50 Daily Builds are for Wimps, *page 74*
- 51 Successful Technical Leading on an OO project, *page 75*
- 52 Notes on the forgotten Art of Software Architecture, *page 75*
- 53 Framework Design and Implementation using Java and UML, *page 76*
- 54 Developing Web Services, *page 76*



## Monday, 4 November

8:30-17:00 Full day

Monday

8:30-17:00

### / Essential Object-oriented Analysis and Design

Jill Aden, EDS, [jill.aden@eds.com](mailto:jill.aden@eds.com)

Joseph Brennan, EDS, [joseph.brennan@eds.com](mailto:joseph.brennan@eds.com)

This tutorial offers quick-paced instruction about object-oriented analysis and design based on practical project experience. Introduction to object-oriented concepts (inheritance, polymorphism, encapsulation, etc.), use cases, class diagrams, sequence diagrams, and patterns. It will provide you with the knowledge and skills to:

- Create use case documents
- Understand object-oriented concepts, terminology and buzzwords
- Identify classes and create class diagrams
- Understand and use the Unified Modeling Language (UML)
- Identify behaviors and create sequence diagrams
- Gain an understanding of what patterns are and review 5 of the GOF patterns
- Recognize other UML diagrams

#### Attendee background

The target audience includes people new to objects and object-oriented concepts. The tutorial is intended for people who have had some exposure to objects but need something else to put all the pieces together.

#### Format

Lecture style, slide presentation.

#### Presenters

*Jill Aden has worked for 17 years with EDS as a Systems Architect and has worked with objects since 1993. She is a member of the OTUG group at University of St. Thomas in St. Paul, MN. At EDS, Aden mentors, consults, and teaches object-oriented concepts internally and externally to clients.*

*Joseph Brennan has worked for 17+ years with EDS as a Systems Architect. He has been working with objects since 1993, is a member of the Twin Cities Java User Group. At EDS, Joseph mentors, consults, and teaches object-oriented concepts internally and externally to clients. Joseph is a Sun Certified Java Instructor, Java Developer, and Java Programmer.*

Monday

8:30-17:00

### 2 Agile Software Development Methodologies: Principles and Practices

Jim Highsmith, Cutter Consortium, [jimh@adaptivesd.com](mailto:jimh@adaptivesd.com)

In our Information Age, competitive advantage comes from speed and flexibility. With shortened product development cycles and rapidly changing business initiatives, the formula for success has been articulated by Tom DeMarco: “Agility: 1, everything else: 0.” Agile Software Development is redefining how software will be delivered in our 21st century economy. There has been a rapidly rising interest in these new approaches to software development such as Extreme Programming, Scrum, Adaptive Software Development, Feature-Driven Development, and Dynamic Systems Development Methodology. Furthermore, scores of organizations have developed their own “lighter” approach to building software.

This tutorial is adapted from the material in Jim Highsmith’s book, “Agile Software Development Ecosystems,” and addresses key questions such as: What are Agile Methodologies? What problem domains do Agile Methodologies address? What are the common principles and practices of the various Agile Methodologies? What are the similarities and differences between the various Agile Methodologies? How would you design your own Agile Methodology?

#### Attendee background

Familiarity with software development or software project management.

#### Format

Lecture and Exercises.

#### Presenter

*Jim Highsmith directs the Cutter Consortium’s Agile Project Management Practice, and author of Agile Software Development Ecosystems, and Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, winner of the Jolt award for 2000. He has over 20 years experience as a consultant, software developer, manager, and writer. In the last ten years, he has worked with both IT organizations and software companies in the US, Europe, Canada, South Africa, Australia, Japan, India, and New Zealand to help them adapt to the accelerated pace of development in increasingly complex, uncertain environments.*

**Monday 8:30-17:00**

### 3 Pattern-Oriented Software Architectures for Networked and Concurrent Applications

Douglas Schmidt, *University of California, Irvine*, [schmidt@uci.edu](mailto:schmidt@uci.edu)

Developing software for distributed systems that effectively utilizes concurrency over high-speed, low-speed, and mobile networks is hard. This tutorial describes how to apply patterns and middleware frameworks to alleviate the complexity of developing concurrent and networked applications. These patterns and framework components have been used successfully by the speaker on production communication software projects at hundreds of commercial companies for telecommunication systems, network management for personal communication systems, electronic medical imaging systems, real-time avionics, distributed interactive simulations, and automated stock trading.

The tutorial illustrates by example how to simplify and enhance the development of applications that use concurrency and networking via the use of:

- Object-oriented design techniques -- such as patterns, layered modularity, and data/control abstraction
- Object-oriented language features -- such as abstract classes, inheritance, dynamic binding, and parameterized types
- Middleware -- such as frameworks for host infrastructure middleware (such as ACE) and distribution middleware (such as Object Request Brokers)
- Advanced operating system mechanisms -- such as event demultiplexing, multi-threading, multi-processing, and explicit dynamic linking

Upon completing this tutorial, attendees will be able to:

- Recognize the inherent and accidental complexities involved with developing concurrent and networked applications.
- Understand precisely how object-oriented techniques and tools can and cannot help to alleviate this complexity.
- Apply key object-oriented design techniques (such as patterns, frameworks, and components) to develop reusable concurrent and networked software artifacts.
- Utilize middleware and OO programming features to create efficient, robust, reusable, and extensible concurrent and networked applications.
- Understand advanced OS capabilities and use them effectively to develop extensible, robust, reusable, and efficient concurrent and networked applications.
- Know where to find additional sources of information on how to successfully apply object-oriented techniques to concurrent and networked applications.

The material presented in this tutorial appears in the books "Pattern-Oriented Software Architecture: Patterns for Concurrent and Distributed Objects", Wiley & Sons, 2000 and the two-volume "C++ Network Programming" series, Addison-Wesley, 2002.

#### Attendee background

This tutorial is intended for software developers who are familiar with general object-oriented design and programming techniques (such as patterns, modularity, and information hiding), fundamental OO programming language features (such as classes, inheritance, dynamic binding, and parameterized types), basic systems programming concepts (such as process/thread management, synchronization, and interprocess communication), and networking terminology (such as client/server architectures and TCP/IP).

#### Format

Slide presentation

#### Presenter

*Dr. Douglas C. Schmidt is an Associate Professor in the Electrical and Computer Engineering department at the University of California, Irvine. His research focuses on patterns, optimization principles, and empirical analyses of object-oriented techniques that facilitate the development of high-performance and real-time distributed object computing middleware on parallel processing platforms running over high-speed networks and embedded system interconnects. Dr. Schmidt is currently the Deputy Director and a Program Manager at the DARPA Information Technology Office (ITO), where he helps set the US IT research and development agenda on autonomous systems, network-centric systems, distributed real-time and embedded systems, and augmented cognition.*

**Monday 8:30-17:00**

### 4 Automated Software Testing: Hands On and Interactive!

Gerard Meszaros, *ClearStream Consulting*, [gerard.meszaros@acm.org](mailto:gerard.meszaros@acm.org)  
Shaun Smith, *ClearStream Consulting*, [shaun.smith@acm.org](mailto:shaun.smith@acm.org)

This tutorial takes the "extreme" out of eXtreme Programming and brings the expertise acquired by the agile software development community to mainstream software developers working on all kinds of projects. The Extreme Programming community has shown the value of writing automated software unit and acceptance tests for increasing quality, reducing development cost and improving agility. Now that we know this is possible, how can we apply these learnings to more traditional software projects? And if you are an XP developer, how can you improve the way you write tests?

The XUnit family of testing frameworks (including JUnit, SUnit, NUnit, VbUnit, etc.) provide an important first step in software test automation. But test automation is much more than just knowing how to program a test in XUnit. The neophyte test-writer is faced with a deluge of questions and issues. What should I be testing? How many tests are enough? What should I focus on while writing my tests? How do I test my tests? How can I make my tests more understandable? How do my tests relate to use cases? How can I make my software easier to test? This tutorial addresses these and many other questions.

Participants will learn important strategies for developing automated unit and acceptance tests for their software and gain valuable experience in applying those strategies. Amongst the topic areas to be presented are: what tests are required, Test First vs. Test Last, key characteristics of automated test software, strategies for achieving those characteristics, test automation patterns, outside-in component specification, back-to-front test coding, the psychological stance required of a test-writer, strategies for making tests repeatable and self-checking, and how automated testing fits with UML.



**Attendee background**

The intended audience is a professional software developer who would like to learn how to test their software more effectively. Participants should be familiar with the object-oriented software development paradigm, and be fluent in one or more object-oriented languages. Familiarity with Java or C# will be useful, but not required. Familiarity with a unit testing framework such as JUnit will be very helpful as the basics of using a testing framework will not be dealt with in detail.

**Format**

The tutorial will be a 40/60 mix of presentation and paper-based exercises. Each topic will be introduced in short presentations by the instructors. The participants will then have the opportunity to apply those concepts in hands-on exercises thus gaining valuable experience in a safe learning environment. Finally, the participants will have an opportunity to discuss what they learned and the advantages of various solutions.

**Presenters**

*Gerard Meszaros built his first unit testing framework in 1996 and has been doing automated unit testing ever since. Along the way, he has become an expert in refactoring of software, refactoring of tests, and design for testability. Gerard has applied automated unit and acceptance testing on projects ranging from full-on eXtreme Programming to traditional waterfall development. He has presented successful tutorials at the past three OOPSLAs, has organized workshops at four previous OOPSLAs, and has presented papers on developing and testing object-oriented software frameworks at two previous OOPSLAs, XP2001 and XP2002.*

*Shaun Smith has been developing object-oriented software systems since 1987. In the last few years, his focus has been on the specification and construction of systems following a “test first” approach as well as the development of frameworks to simplify testing. Shaun has participated in past OOPSLA workshops and has presented papers on framework development and testing at OOPSLA, Smalltalk Solutions, and XP2001.*

Monday

8:30-17:00

**Pragmatic Programming - the First Day**

Dave Thomas, *The Pragmatic Programmers, LLC*,  
dave@pragmaticprogrammer.com

Andy Hunt, *The Pragmatic Programmers, LLC*,  
andy@pragmaticprogrammer.com

It would be nice to think that software development could become a tidy, predictable, orderly business, but as Berard reminds us, “walking on water and developing software from a specification are easy if both are frozen.” Unfortunately, everything changes; nothing is frozen. Our work remains far from being either predictable or tidy. But before you decide to abandon software for something easier (particle physics, perhaps?), we’d like to share with you our secret to developing working software, on time and on budget. The answer is not in tools, languages, or methodologies. Instead, it lies with the skills and attitudes of individual developers, in all of us. Pragmatic Programming isn’t about any particular methodology, development tool set, or implementation language. That’s the easy stuff. Pragmatic Programming is about what’s in your head: your approach to your job, your management of knowledge, and how you go about designing and writing excellent code.

Our tutorial isn’t about ivory-tower theory; it’s practical and down-to-earth, with specific advice based on the bestselling book, “The Pragmatic Programmer.” We’ve also been told it’s a lot of fun. Attendees will come away with a better understanding of the critical role they play as individuals in the success of software development projects. They’ll learn specific techniques for improving themselves, their code, and their teams. People who’ve attended Pragmatic Programming courses find that their work becomes more fun, and their teams jell better.

**Attendee background**

Participants will be developers and technical managers who want to develop better software.

**Format**

A combination of slide-based talks, exercises, and group discussion.

**Presenters**

*Dave Thomas has been developing software since the mid 1970’s. He holds an honors degree in computer science from London University, and is a member of the IEEE Computer Society and the ACM. He ran a successful software company in the United Kingdom before moving to the United States and forming the Pragmatic Programmers with Hunt. Together they authored two books, “The Pragmatic Programmer: From Journeyman to Master” and “Programming Ruby: The Pragmatic Programmer’s Guide”. They are authors of the Manifesto for Agile Software Development, and jointly edit the Construction column for IEEE Software magazine. He is a keen private pilot.*

*Andy Hunt has been developing software since the early 1980’s, in various senior positions at companies large and small, before becoming a consultant. He holds a BS in Information and Computer Science from the Georgia Institute of Technology and is a member of the IEEE Computer Society and the ACM. When not programming, he is an avid jazz musician and woodworker. Andy has authored two books with Dave Thomas: “The Pragmatic Programmer: From Journeyman to Master”, and “Programming Ruby: The Pragmatic Programmer’s Guide”. They are authors of the Manifesto for Agile Software Development, and jointly edit the Construction column for IEEE Software magazine.*

Monday

8:30-17:00

---

## 6 Programming Web Services Applications

---

Francisco Curbera, *IBM T.J. Watson Research Center*,  
[curbera@us.ibm.com](mailto:curbera@us.ibm.com)

Nirmal Mukhi, *IBM T.J. Watson Research Center*, [nmukhi@us.ibm.com](mailto:nmukhi@us.ibm.com)

William Nagy, *IBM T.J. Watson Research Center*, [wnagy@us.ibm.com](mailto:wnagy@us.ibm.com)

Sanjiva Weerawarana, *IBM T.J. Watson Research Center*,  
[sanjiva@us.ibm.com](mailto:sanjiva@us.ibm.com)

Three main specifications currently define Web Services as an open application integration framework: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description, Discovery, and Integration)--and we can expect more to be introduced over the next months. The flexibility of the framework makes it a unique tool to take advantage of both Web and platform specific protocols, such as RMI-IIOP and reliable messaging systems, whose high performance and reliability will not be easily abandoned. However, the growing number of specifications and their flexibility make building a Web service seem more complex than it really is. Getting to know what components and tools are available, which ones to use, and how to use them is still a challenge for developers.

This tutorial will help participants weave a path through the alphabet soup of Web services specifications. Participants will learn the significance of each of the major Web services specifications (SOAP, WSDL, UDDI), the relationship between them, as well as the standard components and tools that are available to take advantage of them. The tutorial will teach the attendees how to build, invoke, and compose Web services, focusing on how to achieve standards-based integration in both Web-based and multi-protocol environments. We will take participants through a full working example, demonstrating the basics of Web services server-side programming in J2EE platforms using standard components. We will also show how applications can access Web services using both XML-based and J2EE protocols using the Web Services Invocation Framework (WSIF).

### Attendee background

This tutorial is targeted at developers and software architects that want to gain a thorough yet practical understanding of how Web services can be used and developed in J2EE platforms. A basic knowledge of Java and familiarity with XML is required.

### Format

The tutorial will combine lecture presentations with support slides and detailed hands-on exercises.

### Presenters

*Francisco Curbera is a Research Staff Member at the IBM T.J. Watson Research Center. He has worked on the definition and implementation of Web services specifications since January of 2000. He was one of the developers of the first Java implementation of SOAP (SOAP4J, later Apache SOAP), and one of the authors of the Web Services Description Language (WSDL) and Web Services Flow Language (WSLF) specifications. He holds a Ph.D. in Computer Science from Columbia University.*

*Nirmal K. Mukhi is a Research Associate at the IBM T. J. Watson Research Center, where he has been working on various Web services technologies since November 2000. He is one of the designers of the Web Services Invocation Framework (WSIF), and the Web Services Gateway. He is also the author of a WSIF tutorial. He holds a Masters degree in Computer Science from Indiana University.*

*William A. Nagy is a Software Engineer at the IBM T.J. Watson Research Center. He is a committer on the Apache SOAP project, and led the integration of Apache SOAP into IBM's WebSphere Application Server. He performed the initial integration of UDDI and the IBM Web Services Toolkit, and was one of the lead authors of the WS-Inspection specification. Most recently he has worked architecting and implementing the IBM Web Services Gateway. He holds a Masters degree in Computer Science from Columbia University.*

*Sanjiva Weerawarana is a Research Staff Member in the Component Systems Group at the IBM T.J. Watson Research Center. He is one of the co-authors of the WSDL and WSFL specifications, and a co-developer of Apache SOAP, the WSIF and the Web Services Gateway. He received a Ph.D. in Computer Science from Purdue University in 1994.*

## 8:30-12:00 Morning

Monday

8:30-12:00

**7 Building Secure OO Systems - A Practical Guide and Overview**Andrew Schneider, BJSS, [as@bjss.co.uk](mailto:as@bjss.co.uk)

Security in computer systems is becoming an increasingly important issue, both to managers and customers. As more business and leisure is conducted via electronic means, the 21st century developer will undoubtedly need knowledge in this area. Despite this, for many systems, security is still an afterthought or even more worryingly, completely ignored. This has been highlighted by a number of high profile security breaches, covered in the media. Security is a large domain, so to be effective, developers and architects need a broad understanding of the techniques and technologies available, along with an appreciation of their application.

This tutorial will provide this information and equip attendees with a framework for handling security issues during design, implementation, and deployment of a system. Other topics will include design patterns, common attacks, common design and implementation mistakes, secure coding and design practices, applications of PKI, J2EE Security and XML based security standards, FIPS 140-1, protecting data in-memory, on disk and in databases, applications of TLS/SSL, hardware security modules, vulnerability analysis and risk assessments. The tutorial will include examples and experiences drawn from real life systems.

**Attendee background**

All attendees wishing to build systems that are more secure and less vulnerable in today's hostile networked environments. Attendees should be developers or architects.

**Format**

Lecture format with real world examples and group discussion.

**Presenter**

*Andy Schneider wears consultant and architect hats for BJSS, a software services organisation. He has been involved in implementing, designing and leading projects utilising object orientated technology since 1990. His primary interests are complex distributed systems, security and technical management. When wearing his consultant hat he finds himself performing presentations and tutorials on a regular basis. When his architect hat is on he has worked on several major secure systems for key financial organisations.*

Monday

8:30-12:00

**8 Introduction to the .NET Framework**Damien Watkins, Project 42, [damien@project42.net](mailto:damien@project42.net)

This tutorial is an introduction to the .NET Framework and the services and facilities it provides. The tutorial's major objective is to describe the architecture of the .NET Framework and demonstrate the use of its services and facilities through examples. Facilities covered will include: the Common Language Runtime, the Base Framework, Language Interoperability, Web Services and ASP.NET.

**Attendee background**

Attendees should have experience programming in an object-oriented language but are not expected to be familiar with the .NET Framework. Knowledge of a distributed/component architecture, such as CORBA or COM, would be beneficial.

**Format**

This tutorial will be lecture based.

**Presenter**

*Dr. Damien Watkins is the founder and Managing Director of Project 42 P/L and is a former lecturer at Monash University. Since 1998 Damien has been engaged with Microsoft Research on the development of the .NET Framework. Component architectures that provide language interoperability have always held a keen interest for Damien, and he has written a paper on this topic, titled "Handling Language Interoperability with the Microsoft .NET Framework." Currently, Damien is writing a book for Addison Wesley on the .NET Framework. Damien taught the world's first university course utilizing the .NET Framework.*



Monday

8:30-12:00

---

## 9 The Art of Writing Use Cases

---

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*,  
[rebecca@wirfs-brock.com](mailto:rebecca@wirfs-brock.com)

John Schwartz, *OHSU*, [schwajoh@ohsu.edu](mailto:schwajoh@ohsu.edu)

Use Cases describe the behavior of a software system from an external usage perspective. There is an art to writing them clearly. Written carefully, use case models convey key usage specifications and can be tied to other requirements. Written poorly, use cases are confusing and ambiguous. This tutorial presents examples of good and bad use case descriptions, and practical techniques for writing three forms of descriptions: narratives, scenarios, and conversations. Narratives are high-level descriptions written from an external perspective. We show how to elaborate high-level descriptions, choosing either a scenario form, which emphasizes sequence, or a conversation, which highlights interactions between a user and the system. Tips for naming use cases, describing policies, errors and exceptions, attaching other important information, describing meaningful pre- and post-conditions, and creating informative glossary entries are also presented. This tutorial will also expose students to techniques for critically reading and revising use cases in various forms.

### Attendee background

Attendees should be looking for practical ways to improve their use-case writing. They should be familiar with writing and reading software requirements and usage descriptions. Attendees could benefit from an introduction to object concepts. However, an object background is not a pre-requisite!

### Format

The tutorial will be presented in a lecture format with slides. Interspersed with each major section will be short, instructor-led exercises that will reinforce the material presented. Students' handouts will include a copy of the slides as well as numerous use case guidelines.

### Presenters

*Rebecca Wirfs-Brock is president of Wirfs-Brock Associates, a firm specializing in the transfer of object analysis and design expertise to organizations and individuals through training, mentoring, and consulting. Rebecca has been involved with object technology since its infancy. She is the inventor of the set of development practices known as Responsibility-Driven Design. From development on the Tektronix implementation of Smalltalk in the early 1980's, through years of development and training experience, she is recognized as one of only a few knowledgeable and influential practitioners of object-oriented design. She spent 17 years as a Software Engineer at Tektronix, where among other accomplishments, she managed the first commercial Smalltalk effort and was the technical lead for the development of Color Smalltalk. She has written articles, lectured and presented tutorials on object analysis, design and management topics, and has been an innovator in object technology since 1984. Together, with Alan McKean she has co-authored a new design book, to be published in 2002 by Addison-Wesley. Recently, she has authored use cases for a telecommunications framework and an online banking system and has mentored teams designs in use case writing, design, architecture and managing incremental, iterative object-technology projects. She practices what she teaches!*

*John Schwartz is a widely known and respected authority on object analysis and design. John has over 15 years of experience developing and managing object-oriented projects in telecommunications, medical, and CAD. Prior to his current position, he was Vice President and Director of Software Architecture of a 120-person telecom information technology group. While with ParcPlace Systems, he influenced the development of the Object Behavior Analysis method pioneered by Adele Goldberg and Kenny Rubin. John was chairman of the OMG's original Object Model Task Force, and developed the model that CORBA is based on. He has contributed to the definition and practical application of Object Behavior Analysis, Responsibility-Driven Analysis and Design Methodologies. He consults on design and methodology to major object-oriented projects. He has conducted over 100 tutorials and classes on object analysis, design, and programming.*

Monday

8:30-12:00

## 10 Programmer's Dozen: Thirteen Practices for Refactoring, Repairing, and Regaining Control of Your Code

Kevlin Henney, *Curbralan Limited*, [kevin@curbralan.com](mailto:kevin@curbralan.com)

There is no shortage of technical wisdom on how to develop clear and robust code, so why is the pained expression on a programmer's face, sifting through all of the source windows trying to make sense of the encrypted code, such a common sight? There are companies whose development culture does not encourage pursuit of knowledge of practice: a penny-wise pound-foolish approach. However, there are many companies and developers that want to push themselves to the state of the art, but seem swamped and bemused by how much state there really is to that art.

This tutorial offers a concrete thirteen-point list of practices (zero through twelve) that can be applied out-of-the-box to reduce code size and complexity, acting as both guidelines for new code and indicators for refactoring. The short list has no ambition to be all that you needed to know about design but were afraid to ask, but it does offer an easily memorable and easily practiced set of guidelines that offer the greatest immediate return on investment - the most bang for your buck or oomph for your Euro.

Attendees will learn a set of specific coding practices that, through principle and example, help them to reduce the amount of source code in their system, to avoid duplication and unnecessary centralization in their code structure, to decouple tangled hierarchies and overly intimate packages, and to write encapsulated interfaces rather than the assembler-esque getter/setter style that is unfortunately so popular in many systems.

### Attendee background

This tutorial is targeted at people who write code for a living, typically using a curly-bracket language (C++, C#, Java, etc.) and are looking for that extra edge that allows them to keep their code live and clean.

### Format

Lecture with slides and additional notes included in the handout material for the session.

### Presenter

*Kevlin Henney is an independent consultant and trainer. The focus of his work is in programming languages, OO, CBD, UML, patterns, and software architecture. He is a regular columnist for Application Development Advisor (UK) and C/C++ Users Journal online, wrote columns in Java Report and C++ Report, and has a forthcoming column in JavaSpektrum (Germany). He is also a member of Hillside Europe, the EuroPLoP program committee, the OT conference program committee, and a regular speaker at conferences in the US and Europe.*

Monday

8:30-12:00

## // Efficient Implementation of Object-Oriented Programming Languages

Craig Chambers, *University of Washington*,  
[chambers@cs.washington.edu](mailto:chambers@cs.washington.edu)

How are object-oriented languages implemented? What features of object-oriented languages are expensive? What compiler optimizations have been developed to make object-oriented languages more efficient? This tutorial addresses these questions. After identifying the main features of object-oriented languages that are challenging to implement efficiently, three classes of implementation techniques are presented. First, run-time system techniques such as virtual function dispatch tables (including complications due to multiple inheritance and virtual inheritance) and inline caches are described. Second, static intra- and interprocedural analyses are discussed that seek to identify at compile-time the possible classes of message receivers, in order to reduce or eliminate the overhead of dynamic binding. Third, ways in which dynamic execution profiles can be exploited to complement static analysis techniques are described. To assess the relative importance of the techniques, empirical measurements of the effectiveness of many of these techniques, as implemented in the Vortex optimizing compiler, are presented for large benchmarks written in Java, C++, and Cecil.

### Attendee background

Attendees should be familiar with the features of object-oriented languages and also with traditional compiler techniques such as procedure inlining and data flow analysis.

### Format

lecture with slides

### Presenter

*Craig Chambers has been researching object-oriented language design and implementation since 1987, with publications in OOPSLA, ECOOP, ISOTAS, PLDI, POPL, PEPM, and TOPLAS on the topic. For his Ph.D. thesis at Stanford, he developed the first efficient implementation of the Self language, using optimizing dynamic compilation. Chambers is currently an Associate Professor of Computer Science & Engineering at the University of Washington, where he designed the Cecil language and co-designed the MultiJava and ArchJava languages, heads the Vortex whole-program optimizing compiler project and the Whirlwind staged compiler project, and co-leads the DyC selective dynamic compilation project.*

13:30-17:00 Afternoon

Monday 13:30-17:00

**12 Card-based User and Use Case Task Modeling**Larry Constantine, *Constantine & Lockwood, Ltd.*,*lconstantine@foruse.com*James Noble, *Victoria University of Wellington, New Zealand*,*kjx@mcs.vuw.ac.nz*Lucy Lockwood, *Constantine & Lockwood, Ltd.*,*llockwood@foruse.com*

Usability is increasingly recognized as a central component of quality in software and Web-based applications, but developers often lack the time and techniques needed for effective user interface design. This tutorial introduces core techniques of agile usage-centered design, a proven approach that complements modern agile methods, such as extreme programming, that have demonstrated their ability to speed design and development but offer too little in the area of software usability and user interface design. The focus in agile usage-centered design is on minimal models that provide maximum payoff in improved designs. The tutorial covers simple but powerful techniques employing ordinary index cards to model the roles users play in relation to a system and the tasks users need to perform within those roles. A variety of techniques are explained and applied, including card storming, role and task inventories, abstract dialogs, role-support analysis, and cooperation clustering of task cases. Task models based on essential use cases will be compared with conventional use cases, scenarios, and user stories, and user roles will be compared with personas and user profiles. The impact of user role and task models on the design of improved user interfaces will be explored.

**Attendee background**

This tutorial is targeted toward practitioners who want to improve their abilities to incorporate user perspectives in their work through usage-centered design. Participants should have some knowledge and experience with use cases. Experience in user interface programming or design is helpful but not required.

**Format**

This tutorial will alternate between short lectures and hands-on application. Hands-on practice will be supported by coaching and consultation from the tutorial leaders and will be followed by class review and discussion of the work and the process.

**Presenters**

*Larry Constantine, Adjunct Professor, Information Technology, University of Technology, Sydney (Australia) and Director of Research and Development, Constantine & Lockwood, Ltd., is a pioneer in software engineering who now focuses on software usability and usage-centered design methods. In a career spanning four decades, he has had over 150 papers published plus 16 books, including the 1999 Jolt Award winner, Software for Use (Addison-Wesley). An award-winning designer (Performance-Centered Design Competition 2001) as well as a respected teacher, he has taught in 17 countries around the world.*

*Dr. James Noble, Senior Lecturer in Computer Science, Victoria University of Wellington (NZ) and Consulting Associate, Constantine & Lockwood, Ltd., is a leading expert on design patterns for OO as well as UI design. He has taught successful seminars and tutorials on usage-centered design and other topics at OOPSLA and elsewhere. With colleagues in the Victoria Object Group, he pioneered streamlined methods for responsibility-driven design with essential (abstract) use cases. Noble is the co-author with Charles Weir of Small Memory*

*Software (Addison-Wesley, 2000) and editor of Prototype-Based Programming (Springer Verlag, 1999) and has numerous published papers to his credit.*

*Lucy Lockwood, President, Constantine & Lockwood, Ltd., is an internationally respected consultant and trainer who draws on nearly 20 years experience in programming and project management. Her practice centers on software usability and technical teamwork, and she has contributed many of the core concepts and techniques in usage-centered design. A top-rated speaker, she has taught around the world and has keynoted major conferences. She is author of more than a dozen published papers and co-author of the award winning book, Software for Use (Addison-Wesley, 1999).*

Monday 13:30-17:00

**13 Developing Java Applications for Mobile Devices**Franz Gruber, *Software Competence Center Hagenberg*,*franz.gruber@scch.at*Markus Irle, *Software Competence Center Hagenberg*,*markus.irle@scch.at*

While Java is very successful and well established on the server and internet application area it has yet to prove itself a viable choice for programming of PDAs and embedded devices. This tutorial focuses on the main aspects, technologies and pitfalls of writing robust, portable and reusable Java components and applications for mobile appliances. The tutorial demonstrates the various different Java specifications, their main features and how they are to be utilized best for the programming of Personal Digital Assistants (PDAs). Design issues as well as platform specific will be treated in the second part of the tutorial. An implemented industrial case study including experience reports will conclude the tutorial.

**Attendee background**

This course is designed for software developers, including architects and programmers. Attendees should have a good knowledge of Java and object-oriented programming, design and analysis methods.

**Format**

Lecture based

**Presenters**

*Franz Gruber: DI at Johannes Kepler University Linz, and Project Manager at the Software Competence Center Hagenberg with main research areas of mobile computing, agent technologies, mobile financial services, and wireless communication technologies. He is a Sun Certified Java Programmer. He was a Developer, with a focus on financial services (Java/Host Integration), at Oberbank, Austria.*

*Markus Irle: CI at Johannes Kepler University Linz, and Member of Scientific Staff at the Software Competence Center Hagenberg with main research areas of e-commerce, m-commerce, and mobile computing.*

Monday 13:30-17:00

**14 Object vs. The Web**

Alan Knight, *Cincom Systems*, [knight@acm.org](mailto:knight@acm.org)  
 Naci Dai, *ObjectLearn*, [nacidai@acm.org](mailto:nacidai@acm.org)

Applying software engineering principles, particularly object-oriented techniques, to the Web is not always easy. Many current Web technologies lend themselves to or even encourage bad practices. For example, scripting and server-page technologies can encourage cut-and-paste reuse, direct-to-database coding, and poor factoring. Component models like COM and EJB seek to construct building blocks for application assembly, but in doing so they sacrifice many of the advantages of objects. XML emphasizes technology-independent reuse and sharing of content, data, and messaging but at the expense of encapsulation and the association of behavior with state, which is central to OO.

However, software engineering and OO techniques are gaining importance in Web development as Web applications become more complex and integrated with traditional server-side applications. To motivate the need for these techniques, we examine some representative Web technologies and the issues they present in naive use. We describe a layered, OO architecture, based on the Model-View-Controller (MVC) pattern, which can overcome these issues to produce large, well-structured systems.

This tutorial surveys current web technologies with an emphasis on OO usage, provides best practices and examples from Java and Smalltalk, discusses myths and truths about components, Web Services, and XML, and describes architecture and development practices that support good practices. Attendees will learn a set of patterns for building well-structured web-based systems which leverage object-oriented design principles.

**Attendee background**

Attendees should have a reasonable understanding of OO development. Experience with web development is helpful, although a basic familiarity with terms is adequate.

**Format**

Presentation

**Presenters**

*Alan Knight works on Smalltalk web tools for Cincom Systems. Prior to that he was chief architect of the TOPLink family of object-relational mapping products with The Object People and WebGain. He has spoken extensively at conferences including OOPSLA, Smalltalk Solutions and Java One, and is co-author of the book "Mastering ENVY/Developer".*

*Naci Dai is the founder of ObjectLearn and an independent mentor and an educator. Prior to that he was with BEA Systems and The Object People Inc. He teaches object technology, Java, design patterns, and distributed computing. He leads and mentors Web development projects. He has a background in applied engineering and computational physics. He has received his Ph.D. from Carleton University. He is a member of the ACM and IEEE.*

Monday 13:30-17:00

**15 Patterns of Enterprise Application Architecture**

Martin Fowler, *ThoughtWorks, Inc.*, [fowler@acm.org](mailto:fowler@acm.org)

The last decade or so has been one of technology churn. We've seen client/server, CORBA, J2EE, COM, .NET and a host of other enterprise platforms appear or fade from view, or both. Keeping up with resulting alphabet soup is a full time job, even without applications to ship.

But amongst all of this churn, some techniques stay relatively constant. So we've been trying to identify these common patterns and taking note of how we use the ideas from one technology and use similar, but not the same designs in others (you can find these at [www.martinfowler.com/isa](http://www.martinfowler.com/isa)).

In this tutorial we'll explore a number of these patterns. We'll touch on various topics including layering, business logic organization, database mapping, web server design, and session state management.

**Attendee background**

Knowledge of the basic topic of developing information systems.

**Presenter**

*Martin Fowler is the Chief Scientist for ThoughtWorks, Inc., a leading custom e-business application and platform development firm. For a decade he was an independent consultant pioneering the use of objects in developing business information systems. He's worked with technologies including Smalltalk, C++, object and relational databases, and EJB with domains including leasing, payroll, derivatives trading and healthcare. He is particularly known for his work in patterns, the UML, lightweight methodologies, and refactoring. He has written four books: Analysis Patterns, Refactoring, the award winning UML Distilled, and Planning Extreme Programming.*



Monday 13:30-17:00

**16 Aspect-Oriented Programming in C++**Detlef Vollmann, *vollmann engineering gmbh, dv@vollmann.ch*

Aspect-Oriented Programming (AOP) has gained a certain momentum in modern software development. The goal of AOP is to provide better modularization by providing techniques for the separation of concerns in software development. A main target of this separation is cross-cutting concerns, which are often distributed among several parts of the system in traditional design.

Most AOP tools provide a weaving mechanism that takes the separately defined cross-cutting concerns and inserts them at the appropriate places into the main code. For that, they take a main language (e.g. Java) and provide a complementary language, which defines the weaving process. A similar approach is possible in C++ only: using native C++ features like namespaces and templates, the C++ compiler can fulfill the weaving job for AOP. But weaving is not always the best solution for AOP. C++ provides a rich set of techniques to separate concerns. One of the most promising ideas is Andrei Alexandrescu's "policy class" approach.

This tutorial presents the weaving approach in C++, discusses some drawbacks of weaving and then presents in-depth the use of policy classes to implement cross-cutting concerns in an AOP way.

**Attendee background**

Participants should have a good working knowledge of ISO C++.

**Format**

Lecture with thorough discussion.

**Presenter**

*Detlef Vollmann has a background of 15 years in software engineering and more than 10 years with object technology. As an independent consultant he supports several Swiss companies with the design of object-oriented systems. Since 1991, he has authored and taught courses in C++, Object-Oriented Technologies, Software Architecture and Distributed Computing for major Swiss companies.*

Monday 13:30-17:00

**17 The Design and Implementation of the Jikes RVM Optimizing Compiler**Michael Hind, *IBM T.J. Watson Research Center, hind@watson.ibm.com*  
David Grove, *IBM T.J. Watson Research Center, groved@us.ibm.com*

The Jikes Research Virtual Machine (RVM) is a software infrastructure designed to execute Java programs typically used in programming language implementation research. The Jikes RVM is available as an open source project. The Jikes RVM provides the academic and research communities with a flexible open testbed to prototype new virtual machine technologies and experiment with various design alternatives. A large number of academic research groups have already adopted it. It runs on AIX and Linux platforms and demonstrates industrial strength performance for many benchmark programs. The Jikes RVM includes state-of-the-art technologies for dynamic compilation, adaptive optimization, garbage collection, thread scheduling, and synchronization.

This tutorial presents an overview of the Jikes RVM optimizing compiler. The first part of the tutorial covers the structure of the compiler, focusing on the requirements, goals, and design of its intermediate representation (IR). The second part of the tutorial covers some of the compiler's extensive set of analyses and optimizations, ranging from simple local analyses to SSA-based flow-sensitive optimizations with type-based heap analysis. The last part covers the integration of the optimizing compiler into the adaptive optimization system, focusing on instrumentation compilation and feedback-directed optimizations.

Specific issues to be covered include: optimizing the memory model, precise exceptions, and dynamic class loading; compiler requirements for runtime support of garbage collection maps, scheduling (yield points), exception tables, line number information; compiler/runtime system cooperation for fast object allocation and runtime services (for example dynamic type checking or invokeinterface); compiler structure for multiple platforms; tracing an interesting method (for example, one of our enumeration loops from the compiler) through key optimizations to illustrate how type analysis, inlining, scalar replacement, plus a set of traditional optimizations work together; etc.

**Attendee background**

Attendees should have a general knowledge of basic compiler analysis and optimization technology and general familiarity with the issues in optimizing object-oriented languages. Detailed knowledge of Java is not required.

**Format**

The tutorial will be lecture based.

**Presenters**

*Michael Hind is a Research Staff Member at IBM's Thomas J. Watson Research Center. He received his Ph.D. from New York University in 1991 and was a professor of computer science at the State University of New York at New Paltz from 1992-1998. He is currently working on the Jikes RVM at IBM, where he is managing the Dynamic Optimization group. His research interests include program analysis, adaptive optimization, and programming languages. Michael was one of two presenters of a full day tutorial at PACT'01 on the Jikes RVM.*

*David Grove is a Research Staff Member at IBM's Thomas J. Watson Research Center. He received his M.S. and Ph.D. from the University of Washington in 1994 and 1998, respectively, and a B.S. from Yale in 1992. He is one of the implementers of Jikes RVM, focusing on the adaptive optimization system, optimizing compiler, and the interface between the virtual machine and the compilers. His research interests include program language design and implementation, and adaptive optimization.*

## Tuesday, 5 November

8:30-17:00 Full day

Tuesday

8:30-17:00

### **18** A Brief Tour of Responsibility-Driven Design

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*,  
rebecca@wirfs-brock.com

Alan McKean, *Wirfs-Brock Associates*, alan@wirfs-brock.com

Responsibility-Driven Design is a way to design that emphasizes behavioral modeling using objects, responsibilities and collaborations. In a responsibility-based model, objects play specific roles and occupy well-known positions in the application architecture. Each object is accountable for a specific portion of the work. They collaborate in clearly defined ways, contracting with each other to fulfill the larger goals of the application. By creating a “community of objects”, assigning specific responsibilities to each, we build a collaborative model of our application.

Objects are more than simple bundles of logic and data... they are service-providers, information-holders, structurers, coordinators, controllers, and interfacers to the outside world! Each must know and do its part! Thinking in these terms enables you to build powerful, flexible applications.

This tutorial, which includes material from our new book, will be an example-based tour of Responsibility-Driven Design. It presents our latest innovations and practical techniques. Topics include: finding and evaluating the qualities of candidate design objects, mapping roles to classes and interfaces, strategies for assigning object responsibilities, deciding on the control style of an application, effective ways to describe collaborations using UML and collaboration stories. Students will have an opportunity to practice techniques with several short exercises.

#### Attendee background

Participants should be familiar with object concepts and be looking for practical techniques, guidelines and a design process that emphasizes modeling the behavioral aspects of a software system.

#### Format

The tutorial will be presented in a lecture format with slides containing notes, guidelines and examples. Interspersed with each major section will be short, instructor-led exercises that reinforce the material presented. Students’ handouts will include a copy of the slides as well as numerous design guidelines.

#### Presenters

*Rebecca Wirfs-Brock is president of Wirfs-Brock Associates, a firm specializing in the transfer of object analysis and design expertise to organizations and individuals through training, mentoring, and consulting. Rebecca has been involved with object technology since its infancy. She is the inventor of the set of development practices known as Responsibility-Driven Design. From development on the Tektronix implementation of Smalltalk in the early 1980’s, through years of development and training experience, she is recognized as one of only a few knowledgeable and influential practitioners of object-oriented design. She spent 17 years as a Software Engineer at Tektronix, where among other accomplishments, she managed the first commercial Smalltalk effort and was the technical lead for the development of Color Smalltalk. She has written articles, lectured and presented tutorials on object analysis, design and management topics, and has been an innovator in object technology since 1984. Together, with Alan McKean she has co-authored a new design book, to be published in 2002 by Addison-Wesley. Recently, she has authored use cases for a telecommunications framework and an online banking system and has mentored teams designs in use case writing, design, architecture and managing incremental, iterative object-technology projects. She practices what she teaches!*

*Alan McKean is Vice President and Director of Educational Services at Wirfs-Brock Associates. Alan is an internationally known instructor of object-oriented software design and development. Alan McKean has devoted most of his career applying principles of design and adult learning to find better ways to communicate technical and design information. A student of R. Buckminster Fuller’s and a graduate of the University of Oregon with a Masters in Computer Science, he specializes in system architecture and object-oriented design and programming. Alan has authored courses including Responsibility-Driven Design, Smalltalk, and Designing with Java and has delivered over a hundred workshops on designing and programming object-oriented software during his 10+ years at Instantiations, Digitalk, and Wirfs-Brock Associates. Alan was a keynote speaker at the OOPSLA Educator’s Symposium in 1995 and 2001. Prior to his training experience, Alan was Director of Audio Engineering at Dynamix, Inc., a computer game company, where he invented and developed a toolset for synchronizing animated images with actors’ recorded voices and a suite of Smalltalk-based tools for managing computer game sound effects and music.*

Tuesday

8:30-17:00

## 19 Concepts of Object-Oriented Programming

Raimund Ege, *Florida International University*, [ege@cs.fiu.edu](mailto:ege@cs.fiu.edu)

This tutorial defines and teaches the basic object-oriented concepts, illustrates their advantages, and introduces the components and features of object-oriented programming languages and development environments. The tutorial enables an attendee to make an informed decision about what language/environment will best serve his/her software development needs.

The tutorial has 2 major parts:

Part 1 discusses in detail all object-oriented concepts and uses UML and Java to illustrate them. The focus will be on a precise non-confusing definition of the core concepts and terminology. Basic object-oriented concepts, such as object, instance, class, interface, attribute, service, message passing, hierarchy, inheritance, polymorphism, late binding, memory management, access specification and packaging.

Part 2 then compares the major object-oriented programming languages: C++, Java, C#, and others. The comparison is done with a double focus: (1) how does the language support and enforce the concepts, and (2) how does the language help software development (to that effect, I have a small case study program, that will be solved in all languages). Whether and how each language supports advanced concepts, like multiple and repeated inheritance, genericity, interfaces, is discussed in detail.

### Attendee background

Attendees are software professionals who are interested in learning the fundamental concepts and advantages of object-oriented programming and how to apply them in a modern software development environment. No previous knowledge of object-oriented concepts is assumed. The attendees should have a fundamental background in computer science and/or computer programming.

### Format

Lecture

### Presenter

*Raimund K. Ege is an Associate Professor of Computer Science at the Florida International University, Miami. He is author of Programming in an Object-Oriented Environment (Academic Press, 1992) and Object-Oriented Programming with C++ (Academic Press, 1994). He is an active researcher in the area of object-oriented concepts, and their application to programming, user interfaces, databases, simulation and software engineering. He has presented numerous successful tutorials at major conferences (OOPSLA, ECOOP, TOOLS). The tutorials were consistently rated highest and won praise from organizers and attendees.*

Tuesday

8:30-17:00

## 20 Web Services and Service Oriented Architectures

Peter Herzum, *Herzum Software*, [herzum@herzumsoftware.com](mailto:herzum@herzumsoftware.com)

This tutorial situates Web services in the architecture, integration, and development of business systems, within large enterprises, product suite development, and across multi-enterprise collaborations. The tutorial introduces the main technical and architectural concepts and patterns specific to Web services and Service Oriented Architectures and an advanced analysis of real-world usage of technical standards such as SOAP, WSDL/WSEL/WSFL, UDDI, and other standards such as ebXML and RosettaNet. Then the tutorial presents the principles required to succeed with developments using Web services, including registers architectures, Web service negotiation, using contracts for specifying Web services, security aspects, importance of ontologies for federations, and more. The tutorial focuses on the architectural and methodological aspects that early adopters have identified as most important for overall scalability and cost-effectiveness, not only in the definition of Web service but in evolving existing architecture, defining new architectures, and managing systems in the presence of Web services. Special importance is given to the relationship between Web services and other integration strategies such as EAI, other development strategies such as components, and to real-world issues such as evolution and configuration management of Web services systems.

Attendees will come away with a real-world perspective of how Web services are being used for component-based development, e-business, system integration, and evolution of federations of business systems, both within large enterprises and across multi-enterprise collaborations, as well as the kind of challenges, and possible solutions, that one can expect from the adoption of Web services.

### Attendee background

Targets architects, technology leaders, and senior developers. Familiarity with distributed system development challenges is required. Knowledge of basic technology standards (XML, SOAP, WSDL, UDDI) is useful but not required.

### Format

Instructor-led, with frequent examples and real-world case-studies and encouraging discussion and interaction with audience.

### Presenter

*Peter (CTO at Herzum Software) has over 15 years of experience in distributed systems architecture, integration, and development. After extensive object-oriented experience, since 1992 he has pioneered project, product, and product-lines development using enterprise components. Since 1999 he has worked with Service Oriented Architectures for large enterprises. Peter is principal author of best-selling book "Business Component Factory", and a respected industry analyst. Chair of the OMG Webservices SIG, he is an internationally recognized authority on large-scale software architecture and development with Web Services and component technology and is a frequent speaker at senior management events and technology conferences world-wide.*



Tuesday

8:30-17:00

## 21 A Language Designer's View of Rotor, Microsoft's "Shared Source" Implementation of the Common Language Infrastructure

David Stutz, *Microsoft Corporation*, davidstu@microsoft.com

Yahya Mirza, *Aurora Borealis Software*, yahya\_mirza@hotmail.com

Recently, Microsoft has introduced the next major evolution of their computing platform, the .NET initiative. The core of this initiative is a language-agnostic runtime system, which is being standardized by ECMA, called the Common Language Infrastructure or the CLI. Finally, the language interoperability and integration solutions promised in the 90's are now becoming a pervasive commercial reality. Furthermore, the multi-vendor adoption of Microsoft's .NET initiative provides a great opportunity for language researchers. This opportunity allows a language researcher to innovate in their particular domain, while interoperating with existing commercial and research oriented language-based solutions.

In early 2002, Microsoft will release Rotor, a "Shared Source" implementation of the CLI available on BSD UNIX as well as the Windows platform. A key technical advantage of the CLI is the ability of its intermediate language to support multiple language paradigms, as opposed to a single rigid object model. For language designers, Rotor can serve as an effective runtime core for experimentation at the language feature level. For compiler and virtual machine researchers, Rotor provides a context for applied research into alternative object representations, method dispatch, garbage collectors, JIT compilers, etc. Our goal is to provide an in-depth exploration into Rotor.

Not only will attendees learn about the inner workings of the Rotor implementation, they will also learn about alternative design options and their ramifications which led to the current architecture of the CLI. The approach to describing Rotor will be based on identifying the patterns and idioms embodied within Rotor, as well as describing their implementation details. Additionally, we plan to illustrate similarities and differences with other runtime systems including the Squeak and Java virtual machines. Another key objective of this tutorial will be to illustrate approaches to extending and modifying Rotor in new directions. These approaches to language feature extensions will include the description of language features based on runtime or meta-level frameworks, as well as approaches to extending the underlying intermediate language. To illustrate the issues involved in research extensions to the intermediate language, Microsoft Cambridge's ILX extensions will be dissected and explained in detail. Finally, language interoperability issues in the context of language feature design will be enumerated. As a working example, how C# layers over the Common Language Subset or the CLI object model will be explained in detail.

### Attendee background

This tutorial is aimed at professionals wanting a deeper understanding of Microsoft's CLI and its comparison with other runtime systems. Graduate students interested in language design, compiler implementation, or virtual machine research, as well as researchers requiring a host environment to construct a domain specific language would be ideal candidates.

### Format

This tutorial will be presentation based using slides.

### Presenters

*David Stutz is currently working on the team that is implementing the Microsoft Shared Source CLI. He is also well known for his kibitzing on the design of peer-to-peer and distributed computing infrastructures. During his tenure at Microsoft and Microsoft Research, he has participated in designing programming languages, component technologies, operating systems, developer tools, and a whole lot of software plumbing. He is also an accomplished early music performer and a wine grape farmer.*

*Mr. Mirza has had the pleasure of bugging David Stutz and Ralph Johnson about applied object technologies since 1995. In 1996, Mr. Mirza bugged Ralph so much that he was invited to study under Ralph at the UIUC. Mr. Mirza's passion for components originated from his work in the aerospace industry, to quickly generate models of single-stage-to-orbit design concepts. After transitioning into the computer animation industry, Mr. Mirza has been developing a component-based infrastructure for an animated production. Mr. Mirza continues to perform original guitar and piano music around Seattle until he makes it in the software industry.*

8:30-12:00 Morning

Tuesday

8:30-12:00

## 22 Extreme Programming, A Simulation

Joshua Kerievsky, *Industrial Logic, Inc.*, [joshua@industriallogic.com](mailto:joshua@industriallogic.com)

Rob Mee, *Pivotal Computer Systems*, [robmee@hotmail.com](mailto:robmee@hotmail.com)

Interactive learning designers and military organizations have known for years that the best way to learn a process is to experience a comprehensive and memorable simulation of it, followed by a thorough debrief. In this half-day tutorial, both non-technical folks (including managers) and technical folks (including programmers and QA engineers) will get to experience a thorough simulation of Extreme Programming (XP).

The simulation involves the completion of a non-technical project, in which participants, playing the roles of XP Customer or Programmer, experience all of the XP practices, including Iteration Planning, Automated Testing, Test-First Development, Continuous Integration, Collective-Ownership and Pairing. The tutorial's instructors - two experienced XP Coaches -- will play the role of Coach. During the simulation debrief, students will do an Iteration Retrospective and learn how their simulated experiences map to real-world XP practices and projects.

Participation in this tutorial requires a laptop loaded with Microsoft Word and Excel, an ethernet card and ethernet cable (RJ-45), and power supply.

### Attendee background

This tutorial is open to anyone who is interested in learning about Extreme Programming.

### Format

Highly interactive session.

### Presenters

*Joshua Kerievsky is a software development coach and programmer. After programming on Wall Street for nearly 10 years, in 1995 he founded San Francisco Bay Area-based Industrial Logic (<http://industriallogic.com>), a company that specializes in Extreme Programming (XP). Since 1999, Joshua has been programming and coaching on small, large and distributed XP projects and teaching XP to people throughout the world. He regularly speaks about XP, has authored numerous articles, simulations and games about XP and patterns and is working on the forthcoming book Refactoring to Patterns (<http://industriallogic.com/xp/refactoring/>).*

*Rob Mee has been a consultant since 1986 and has been involved with XP since 1998, coaching a variety of projects in both IT shops and product companies. During 2000 and 2001 he served as Director of Engineering and XP Coach at Evant, a software vendor in San Francisco. He is a regular lecturer on XP, and has spoken about XP to large corporations, workshops and venture capital firms. He has been featured in interviews by a number of magazines including Forbes and InformationWeek. Rob's projects, including his coaching work at Evant, are often cited by agile methodology gurus such as Kent Beck and Alistair Cockburn as examples of real XP in action.*

Tuesday

8:30-12:00

## 23 Designing an Agile Methodology

Alistair Cockburn, *Cockburn and Associates*, [acockburn@aol.com](mailto:acockburn@aol.com)

A methodology is a social construction that includes the roles, skills, teaming, activities, techniques, deliverables, standards, habits and culture of the organization as it develops software. The first part of the tutorial introduces language and constructs needed to evaluate, compare and construct methodologies. These include precision, accuracy, tolerance, relevance, and scale, along with the nine basic elements of a methodology. Definition and principles for designing agile methodologies are given, along with three examples of effective agile, increasingly lightweight (real) methodologies are given [Crystal Orange, Crystal Clear, Extreme Programming], along with commentary on the social setting for each. The tutorial examines the conditions suited to shifting from a lighter to a heavier methodology and the penalty for doing so. Considerations about success and failure in affecting culture are visited again at the end. Learn to identify and diagnose the parts of your organization's methodology, and learn ways to make it more effective. Attendees should have significant software team experience, preferably but not necessarily OO, and must have used at least one methodology and thought about others.

### Attendee background

Experienced developers, team leaders, methodologists and technology selectors trying to choose or design a methodology for their organization.

### Format

Lecture

### Presenter

*Alistair Cockburn, founder of Humans and Technology and Cockburn and Associates, has been known for his work on light and agile methodologies for a decade. He was special advisor to the Central Bank of Norway, and the designer of the IBM Consulting Group's first OO development methodology. He is the author of "Surviving OO Projects," "Writing Effective Use Cases," and "Agile Software Development." He is an expert on use cases, object-oriented design, project management, and software methodologies. He has been the technical design coach and process consultant on projects ranging in size from 3 to 90 people. Materials that support Alistair's workshops can be found at <http://members.aol.com/acockburn> and <http://crystalmethodologies.org>.*

Tuesday

8:30-12:00

## 24 J2EE for Enterprise Application Integration and e-business Integration

Matjaz B. Juric, *University of Maribor*, [matjaz.juric@uni-mb.si](mailto:matjaz.juric@uni-mb.si)

Enterprise Application Integration (EAI), although not a buzzword anymore, is still very important for each company. Among others, it is the prerequisite for successful B2B integration. This tutorial gives answers to the following questions: How suitable is J2EE for EAI? Which steps should we do for successful integration and which patterns should we use? How should we design the integration architecture? How should we wrap existing applications? When should we use virtual components? How can we achieve transactional and security integration?

The tutorial also discusses the suitability of J2EE technologies for integration and provides practical examples how to achieve interoperability between CORBA and RMI, different MOM (message oriented middleware) systems, transactional systems (X/Open DTP, CORBA OTS), security mechanisms, JNDI, EIS systems and Windows platform. The following will be covered:

- Data level integration (JDBC, JDO, XML)
- Application interface and business method integration (CORBA, RMI-IIOP, EJB, Java Connector Architecture, COM Bridges)
- Presentation integration (JSP/Servlets, XML/XSLT, Portal Servers)

Finally, the tutorial will show how to use the integration architecture for e-business and how to develop Web Services on top. We will demonstrate why successful EAI is the prerequisite for effective Web Services and why tying Web Services directly to existing applications will not work in long term.

### Attendee background

Basic familiarity with Java 2 Enterprise Edition (J2EE)

### Format

Lecture based

### Presenter

*Matjaz B. Juric, Ph.D. is an Assistant Professor at University of Maribor. He is the lead author of Professional J2EE EAI and coauthor of Professional EJB, both books published by Wrox Press. He has published a chapter in More Java Gems (Cambridge University Press) and several original scientific papers in journals such as ACM SigPlan Notices and Information and Software Technology Journal. He has also published in Java Report, Java Developers Journal, JavaWorld, and Web Services Journal and presented on conferences such as OOPSLA, Java Development, ICPADS, PDCS, Wrox Conferences, etc. He has been involved in performance analysis and optimization by the development of RMI-IIOP, an integral part of Java 2 platform, in cooperation with IBM Java Technology Centre. He is a reviewer for John Wiley & Sons, Wrox Press, several journals and conferences.*

Tuesday

8:30-12:00

## 25 Introduction to Concurrent Programming in Java

David Holmes, *DLTeCH Pty Ltd*, [dholmes@dltech.com.au](mailto:dholmes@dltech.com.au)

Doug Lea, *State University of New York (SUNY) at Oswego*, [dl@cs.oswego.edu](mailto:dl@cs.oswego.edu)

Concurrent programming has mostly been the domain of systems programmers rather than application developers, but Java's support of concurrency has enticed many to try their hand at concurrent applications. However concurrent programming poses many traps for the unwary.

This tutorial demonstrates various design patterns and techniques for constructing concurrent applications in Java and for managing that concurrency. On the language side we look at Java's mechanisms to support concurrent programming. On the design side we look at object structures and design rules that can successfully resolve the competing forces (safety, liveness, efficiency, coordination, reusability) present in concurrent software design problems.

### Attendee background

This tutorial targets anyone involved, or planning to get involved, in the development of concurrent object-oriented applications. It is assumed that the attendee is familiar with basic OO concepts and has a working knowledge of the Java programming language.

### Format

Lecture based

### Presenters

*David Holmes is Director and Chief Scientist at DLTeCH Pty Ltd, located in Brisbane, Australia. His work with Java technology has focused on concurrency and synchronization support in the language and virtual machine and he is currently working on a real-time Java virtual machine. David has presented tutorials on concurrent Java programming and design at numerous international object-oriented programming conferences. He is a co-author of the book "The Java Programming Language" (third edition). David completed his Ph.D. at Macquarie University, Sydney, in 1999.*

*Doug Lea is a professor of Computer Science at the State University of New York at Oswego. He is author of the Java Series book "Concurrent Programming in Java: Design Principles and Patterns," co-author of the book "Object-Oriented System Development," and the author of several widely used software packages, as well as articles and reports on object-oriented software development.*

Tuesday 8:30-12:00

**26 Reflection in Java**Ira Forman, IBM, [formani@us.ibm.com](mailto:formani@us.ibm.com)Nate Forman, Liaison Technology, [nforman@austin.rr.com](mailto:nforman@austin.rr.com)

The use of reflection is an important technique for improving productivity. Reflection facilitates development of programs that are easily adapted to requirement changes. With reflection one can develop software engineering tools that examine or produce code. Reflection facilitates testing and problem determination by facilitating the automation of more tedious tasks. In general, reflection improves the flexibility, extensibility, and reusability of one's code.

The Java programming language (version 1.4) contains a highly effective reflection facility. The tutorial explains the concept of reflection, the Java metaobjects (including both introspective and intercessional interfaces), the proxy class, and dynamic compilation and class loading. The limits of Java reflection are addressed in the context of what reflection is capable of in general. In addition, the tutorial demonstrates the efficacy of the Java reflection facility for solving practical problems. Such problems include: program/application testing, generation of code, inspection of code, and use of dynamic class loading in a framework for application extension. The last topic covered will be the performance impact of using reflection.

**Attendee background**

An attendee must be a competent Java programmer.

**Format**

Lecture from slides

**Presenters**

*Dr. Ira R. Forman works for IBM in Austin. As a member of IBM's Object Technology Products Group, which produced the SOMobjects Toolkit, he worked on the SOM Metaclass Framework. He started working in the area of object-oriented programming in 1984, when he worked at ITT Programming Technology Center. Forman received his Ph.D. in Computer Science from the University of Maryland, where he studied under Harlan Mills. Forman's specialties are object-oriented programming, distributed systems, and object composition. He is the coauthor of two books: "Interacting Processes: A Multiparty Approach to Coordinated Distributed Programming" and "Putting Metaclasses to Work: A New Dimension in Object-Oriented Programming."*

*Nate Forman works for Liaison Technology where he designs and programs application frameworks for their products. His specialties are patterns and object-oriented programming. Forman holds a MSE in Software Engineering from the University of Texas at Austin and a BS in Computer Science from the College of Engineering at Cornell University.*

Tuesday 8:30-12:00

**27 Patterns at Work**

Frank Buschmann, Siemens AG, Corporate Technology,

[Frank.Buschmann@mchp.siemens.de](mailto:Frank.Buschmann@mchp.siemens.de)

In this tutorial we present in detail a part of a concrete real-world system and how it is designed with patterns: the representation of physical storage in a warehouse management system as well as the client interface to this subsystem. Step by step we will re-play the process of the system's construction. We discuss the design problems that occur, present the patterns that could help solve these problems, discuss design alternatives and show how we actually applied the patterns we selected. By this we will see how the design of the system slowly grows and evolves towards the final architecture. We will also see and discuss how patterns are applied in practice and how they help building high-quality software with predictable properties. The tutorial concludes with a summary of our experiences from several projects in which we applied patterns: what worked, what could be improved, and what did we learn.

**Attendee background**

Sound knowledge in Object Technology. Basic Knowledge of UML Notation. Basic Knowledge of the pattern concept.

**Format**

Slides, tutorial style but with much interaction with the audience.

**Presenter**

*Frank Buschmann is senior principal engineer at Siemens Corporate Technology in Munich, Germany. His interests include Object Technology, Frameworks and Patterns. Frank has been involved in many software development projects. He is leading Siemens' pattern research activities. Frank is co-author of "Pattern-Oriented Software Architecture -- A System of Patterns" and "Pattern-Oriented Software Architecture -- Patterns for Concurrent and Networked Objects"*

Tuesday

8:30-12:00

## 28 Use Cases, Usability Requirements and User Interfaces

Jim Heumann, *Rational Software*, [jheumann@rational.com](mailto:jheumann@rational.com)

Use cases are a way to capture the functional requirements of a software system. To keep them simple and easy-to-read they purposely don't address either usability requirements or the user interface with which users will interact. Both of these things, however, are important to the success of software that has significant user interaction. This tutorial will introduce a technique, based on the Rational Unified Process, called User Experience Use Case Analysis that uses use cases as input, to produce a conceptual design of the UI and to specify its usability requirements. The main outputs of User Experience Use Case Analysis are Use Case Storyboards, which define the system's conceptual screens, screen content and navigation paths. Attendees at this tutorial will get a comprehensive introduction how GUI's fit with use cases and the techniques used to create a conceptual user interface design and specify usability requirements.

### Attendee background

Attendees should be familiar with use cases, object orientation and UML at an introductory level.

### Format

Lecture/presentation interspersed with discussion of the examples and an in-class exercise.

### Presenter

*Jim Heumann is Requirements Management Evangelist for Rational Software. Throughout his twenty year career as a designer, implementer, maintainer, consultant and manager he has been able to experience, first-hand, most of the challenges we all face day-to-day as software professionals. He has worked on projects in artificial intelligence, cell phone fraud detection, user interface design, visual data analysis, currency trading, image processing and wind profilers, among other. Jim has delivered over fifty workshops, tutorials, training classes and conference presentations over the last ten years on a variety of subjects including Requirements Management with Use Cases, Object-Oriented Analysis and Design, Visual Data Analysis, UML, and Use Cases & Test Cases. He holds an MS in Management Information Systems from the University of Arizona.*

Tuesday

8:30-12:00

## 29 .NET Primer with C#

Michael Stal, *SIEMENS AG Corporate Technology*,  
[Michael.Stal@mchp.siemens.de](mailto:Michael.Stal@mchp.siemens.de)

In October 2000 Microsoft announced the .NET infrastructure as the fundamental layer of all upcoming products. Similar to Java, Microsoft .NET introduces a virtual execution system where applications are executed within a virtual engine. In contrast to Java, however, Microsoft .NET is designed for multi-language support. In addition to the runtime environment, the .NET framework introduces a whole set of libraries for building distributed systems, desktop clients, or web applications. The runtime, parts of the libraries, and the new programming language C# are standardized at ECMA and ISO so that .NET will also be available for non-Windows systems such as Linux. Thus, .NET seems to be a very interesting approach for future software development and a viable alternative to Java. It is the goal of the tutorial to give a detailed overview on C# and .NET. Additional demos will illustrate fundamental issues.

### Attendee background

Familiarity with object-oriented technologies. Knowledge in Java is helpful.

### Format

Slide presentation with small demos.

### Presenter

*Michael Stal is Senior Principal Engineer at Siemens responsible for research on Middleware and Application Integration. Michael focuses on Distributed Objects & Components, Software Architecture, and Web technologies. He is co-author of Pattern-Oriented Software Architecture Volume I&II, Siemens representative at the OMG, former member of the C++ X3J16 standardization working group, and editor-in-chief of Java Spektrum.*



13:30-17:00 Afternoon

Tuesday

13:30-17:00

## 30 eXtreme Programming & Web Development

Joseph Pelrine, *MetaProg GmbH*, [jpelrine@acm.org](mailto:jpelrine@acm.org)

Claudia Chiecchi, *Independent*, [c.chiecchi@digital-prepress.ch](mailto:c.chiecchi@digital-prepress.ch)

eXtreme Programming provides software developers with techniques for dealing with customers and managers. What happens, though, in a Web-based (or other front-end-heavy) project, where you have to deal with graphics artists, web designers, and other creative types? If you've ever lived through a project that was fraught with confusion, mixed messages, and ever-changing parameters, you know what we mean. In this tutorial, we'll address the issues involved in doing graphically intensive Web development, and will present you with a framework for a cohesive Web workflow plan that will help you save time, money and headaches.

Participants will get a better idea of the language the "other side" speaks, the tools and techniques they use, and how to best go about participating in an XP-based Web design workflow process. We will take a look at the principles, techniques, and tools of graphic design, discuss Web design tools, analyze the communications process between designer, developer and customer (especially the problems), and propose a series of steps for stabilizing and increasing the quality of the development process.

### Attendee background

Participants should have some experience in the basics of eXtreme Programming. This tutorial is aimed at software developers as well as graphics and web designers, and also anyone responsible for leading a multi-disciplinary development effort. Although aimed at web development, the techniques are applicable to any UI- and graphic-intensive application.

### Format

This tutorial will be lecture based, with numerous practical demonstrations of tools and techniques.

### Presenters

*Joseph Pelrine is C\*O of MetaProg, a company devoted to increasing the quality of software and its development process. He has had a successful career as software developer, project manager and consultant, and has spoken about it at such diverse places as IBM, OOPSLA and the Chaos Computer Club. Having survived working with Kent Beck, he currently works with Dave Simmons on and in SmallScript when he's not helping his clients solve their problems.*

*Claudia Chiecchi is an experienced graphics designer, specializing in packaging design and corporate identity. An honors graduate of the graphics master class at the Basel School of Art and Design, and protégé of internationally renowned graphic designer Wolfgang Weingart, her clients include such companies as Adobe, Tissot, Weleda, Actelion, and numerous others.*

Tuesday

13:30-17:00

## 31 Patterns for Writing Effective Use Cases

Alistair Cockburn, *Humans and Technology*, [alistair.cockburn@acm.org](mailto:alistair.cockburn@acm.org)

Steve Adolph, *WSA Consulting Inc.*, [steve@wsaconsulting.com](mailto:steve@wsaconsulting.com)

Paul Bramble, *Emperative, Inc.*, [pbram47077@aol.com](mailto:pbram47077@aol.com)

Use Cases are a wonderfully simple concept: describe a system's functional requirements by telling stories about how using it delivers value to its actors. Yet, for such a simple concept, good use cases can be hard to write. This tutorial is a chance for practiced use case writers to gather and ask the hard questions. How should we structure use cases? What should be the scope of a use case? What level of detail should a use case express? What expertise is required in the use case writing team? How do we provide different levels of detail for different people? How do we keep use case writers from infringing on design? Where do we put the UI design, data descriptions and all the other requirements?

Usually, the answers to these questions begin with the very unsatisfying "Well it depends...", or involves using a monolithic process that offers guidance, but creates mounds of unnecessary work.

This tutorial presents a pattern language that allows the attendees to tailor the answers to these questions to their own situation and development culture. It captures the knowledge and experience of successful use case developers in a way that it can be easily re-used by others, and serves as the basis of a vocabulary describing the properties of quality use cases.

### Attendee background

Attendees must have written some use cases and be familiar with basic use case concepts.

### Format

The tutorial is structured as a lecture with slides, and an open question-and-answer session. The lecture introduces a set of patterns designed to help the attendees evaluate existing use cases as well as write better ones. The open question-and-answer section allows the attendees to ask questions currently plaguing them at work, and even trade answers and experiences amongst themselves.

### Presenters

*Alistair Cockburn is a highly regarded instructor and is known as one of the premier experts on use cases. His book "Writing Effective Use Cases" set the standard in the area and was nominated for Software Development's Jolt book award in 2001. Alistair has taught use case writing since 1994, and has also acted as consultant on project management, object-oriented design, and methodology to the Central Bank of Norway, the IBM Consulting Group, and the First Rand Bank of South Africa. Materials that support his workshops can be found at <http://members.aol.com/acockburn>, <http://crystallmethodologies.org> and <http://usecases.org>.*

*Steve Adolph is a senior consultant with WSA Consulting Inc. where he specializes in software process improvement. He has seventeen years of industrial experience developing and managing software development projects in the telecommunications, railway signaling, and graphic arts industry. An exciting and enthusiastic speaker, Mr. Adolph has spoken at numerous seminars and workshops on the topics of software project management object-oriented analysis and design, and patterns. He is co-author of the forthcoming book "Patterns for Effective Use Cases".*

*Paul Bramble is a Senior Software Engineer with Emperative, Inc., and specializes in Object-Oriented Software Development. He has been developing software in the telecommunication, avionics and computer manufacturing industries for over 20 years. He has been using, researching, and writing about use cases since 1994. He led a workshop on "Use Case Patterns" at OOPSLA 98, and is co-author of the forthcoming book "Patterns for Effective Use Cases".*



Tuesday

13:30-17:00

## 32 Dungeons and Patterns!

Steve Metsker, *Capital One*, [Steve.Metsker@acm.org](mailto:Steve.Metsker@acm.org)

William Wake, *Independent Consultant*, [William.Wake@acm.org](mailto:William.Wake@acm.org)

"Dungeons and Patterns!" is a hands-on tutorial for exploring and learning about design patterns. Learning design patterns will help you become a more powerful object-oriented developer. Unfortunately, a single reading of "Design Patterns" won't magically implant design pattern skills in your brain. You have to learn patterns by doing, which means that you have to start applying patterns before you can start applying patterns--a monstrous dilemma! The solution is to practice patterns in a playful setting where slip-ups are profitable and painless.

In this tutorial you will encounter dungeon devices, creatures, and settings that serve as metaphors for design patterns. Your mission, with your table-mates, is to determine which pattern is at play, and then to explain the pattern to an Ahobbit that will accompany your party. As your Ahobbit gains enlightenment, you will find magical doors unlocking that let you delve deeper and deeper into the patterns dungeon. Learning patterns by playing will produce long-term memories, greatly strengthening your skill at recognizing and applying design patterns.

### Attendee background

Attendees should have at least a little knowledge of design patterns, specifically those in the book "Design Patterns." Experience with teaching and/or experience with role playing games is a plus, but not necessary.

### Format

We start with a few slides for orientation, and then we choose Pattern Masters to act as leaders at each table. The Pattern Master has a guide to the dungeon, and the other attendees adventure through the dungeon. After the break and at the end we have a few more expository, informational, and summary slides. Most of the time is devoted to learning and play in the dungeon.

### Presenters

*Steve Metsker is a researcher and author who explores and writes about ways to expand the abilities of developers. Steve is the author of "Building Parsers in Java" and the "Design Patterns Java Workbook."*

*William Wake is an independent consultant and coach. He's the inventor of the XP Programmer's Cube, and the author of "Extreme Programming Explored" and the forthcoming "Refactoring Workbook."*

Tuesday

13:30-17:00

## 33 Concurrency Utilities: Tools for Managing Multithreading

Doug Lea, *State University of New York (SUNY) at Oswego*, [dl@cs.oswego.edu](mailto:dl@cs.oswego.edu)

David Holmes, *DLTeCH Pty. Ltd.*, [dholmes@dltech.com.au](mailto:dholmes@dltech.com.au)

Semaphores and mutexes, latches and gates, pools and executors - these are all tools for managing threads within our applications. Each tool has its own properties and purpose and when used correctly can greatly simplify concurrent program design.

The Java programming language has turned a generation of applications programmers into concurrent programmers through its support of multithreading. But concurrent programming poses many traps for the unwary. Designing concurrent programs that can successfully resolve the competing forces of safety, liveness, efficiency, coordination, and reusability, is a task that can be greatly aided through the use of a good set of tools - a good concurrency utility library.

This tutorial presents the components of a proposed concurrency utility library for the Java platform - JSR-166. We will look at the purpose and properties of each tool and how it fits into the concurrent design puzzle. We will also look at proposed extensions to the Java concurrency primitives.

### Attendee background

This tutorial targets anyone involved, or planning to get involved, in the development of concurrent object-oriented applications. It is assumed that the attendee is familiar with basic OO concepts and has a working knowledge of the Java programming language. Familiarity with Java's threading and synchronization mechanisms is desirable, but not essential.

### Format

Lecture based

### Presenters

*Doug Lea is a professor of Computer Science at the State University of New York at Oswego. He is author of the Java Series book "Concurrent Programming in Java: Design Principles and Patterns," co-author of the book "Object-Oriented System Development," and the author of several widely used software packages, as well as articles and reports on object-oriented software development.*

*David Holmes is Director and Chief Scientist at DLTeCH Pty. Ltd., located in Brisbane, Australia. His work with Java technology has focused on concurrency and synchronization support in the language and virtual machine and he is currently working on a real-time Java virtual machine. David has presented tutorials on concurrent Java programming and design at numerous international object-oriented programming conferences. He is a co-author of the book "The Java Programming Language" (third edition). David completed his Ph.D. at Macquarie University, Sydney, in 1999.*

Tuesday

13:30-17:00

## 34 C++ Idioms

Jan Christiaan van Winkel, *AT Computing*, [jc@ATComputing.nl](mailto:jc@ATComputing.nl)

This tutorial will address non-trivial C++ programming constructs that experienced programmers use frequently. These constructs are too small to be called patterns, but they are part of the language skills of the proficient C++ programmer. Therefore we call them idioms. Several idioms will be discussed. For example, one idiom to be presented is “traits.” Traits classes are used frequently in the C++ standard. Some libraries are built around traits classes, for example the Boost library ([www.boost.org](http://www.boost.org)). Using traits, it is possible to get hold of information about a type at compile-time that will then influence behavior at runtime. Another idiom to be discussed is “intermediate objects.” Intermediate objects are often used as proxy objects to write something in C++ that the syntax disallows, such as “a[1][2]” where a is of class type. They exist only in the expression in which they are used. Several other idioms will be covered.

### Attendee background

The attendees are expected to know C++.

### Format

This tutorial will be lecture based

### Presenter

*JC van Winkel has a B.S. and an M.S. in computer science (the M.S. from the Vrije Universiteit Amsterdam). He works at AT Computing, a small courseware and consulting firm in Nijmegen, the Netherlands. There he teaches UNIX and UNIX-related subjects, including C++. Except for 1995, J.C. van Winkel has presented tutorials at all OOPSLAs since 1993.*

Tuesday

13:30-17:00

## 35 What's Happening Inside Your EJB Application Server?

Scott Crawford, *Independent*, [scott@scottcrawford.co.uk](mailto:scott@scottcrawford.co.uk)

This session goes “under the hood” of some of the most popular Enterprise JavaBeans application servers and explores what’s really going on in there. Some simple EJB programming techniques will be presented which will allow you to study your EJB container, and to appreciate the strategies it uses to manage your components. Topics such as instance pooling, passivation, and caching will be explored with reference to specific J2EE servers.

In addition to learning some coding tricks that will help you with your own studies, the session will include test results from several of the most popular application servers, and will point out some of the most important ways in which they differ.

### Attendee background

Attendees should be familiar with basic EJB concepts such as session, entity, and message-driven beans. Ideally they should have some hands-on experience with at least one application server.

### Format

This tutorial will be primarily lecture format, although attendees will be challenged to discuss and vote on certain specific questions.

### Presenter

*Scott Crawford was a pioneer with Enterprise JavaBeans (tm) technology and remains a leading practitioner. He is known as the creator of both EJB Roleplay, the game for teaching EJB lifecycle, and the BeanBank demonstration, which was an early working example of EJB. In recent years he has presented multiple technical sessions at both JavaOne and the OT/Oxford conferences. A veteran of IBM's Java Technology Centre, today he is an independent consultant working as an architect on a large J2EE project in London.*

Tuesday

13:30-17:00

## 36 Patterns and Application Experiences for Real-time Object Request Brokers

Douglas Schmidt, *University of California, Irvine*, [schmidt@uci.edu](mailto:schmidt@uci.edu)

Distributed, real-time, and embedded (DRE) applications are becoming increasingly widespread and important. Examples of DRE applications include telecommunication networks, e.g., wireless phone services, tele-medicine, e.g., remote surgery, manufacturing process automation, e.g., hot rolling mills, and defense systems, e.g., avionics mission computing systems. Although there are many types of DRE applications, they have one thing in common: the right answer delivered too late becomes the wrong answer.

This tutorial presents design challenges and solutions encountered when building DRE applications using Real-time CORBA middleware. The focus of the tutorial is on Real-time CORBA capabilities that allow DRE applications to configure and control the following resources:

- Processor resources via thread pools, priority mechanisms and intraprocess mutexes
- Communication resources via protocol properties and explicit bindings with non-multiplexed connections and
- Memory resources, via buffering requests in queues and bounding the size of thread pools.

The examples in the tutorial are based on a widely used open-source object request broker (ORB) called TAO, which is a fully-functional Real-time CORBA ORB that illustrates key design patterns and OO programming idioms for developing DRE applications and middleware. The material presented in this tutorial appears in the books “Pattern-Oriented Software Architecture: Patterns for Concurrent and Distributed Objects”, Wiley & Sons, 2000 and a series of columns published in C/C++ Users Journal with Steve Vinoski.

### Attendee background

The tutorial is intended for software developers who are familiar with general object-oriented design and programming techniques (such as design patterns, modularity, and information hiding), fundamental OO programming language features (such as classes, inheritance, dynamic binding, and parameterized types), basic systems programming concepts (such as process/thread management, synchronization, and interprocess communication), and networking terminology (such as client/server architectures and TCP/IP). Some knowledge of CORBA is useful, but not essential.

### Format

Slide presentation.

### Presenter

*Dr. Douglas C. Schmidt is an Associate Professor in the Electrical and Computer Engineering department at the University of California, Irvine. His research focuses on patterns, optimization principles, and empirical analyses of object-oriented techniques that facilitate the development of high-performance and real-time distributed object computing middleware on parallel processing platforms running over high-speed networks and embedded system interconnects. Dr. Schmidt is currently the Deputy Director and a Program Manager at the DARPA Information Technology Office (ITO), where he helps set the US IT research and development agenda on autonomous systems, network-centric systems, distributed real-time and embedded systems, and augmented cognition.*

Tuesday

13:30-17:00

## 37 Objects, XML and Databases

Akmal Chaudhri, *IBM developerWorks*, [akmal@soi.city.ac.uk](mailto:akmal@soi.city.ac.uk)

Richard Edwards, *School of Informatics, University of Wales, Bangor*, [rich@informatics.bangor.ac.uk](mailto:rich@informatics.bangor.ac.uk)

Managing XML documents is problematic when document collections grow. How do we successfully store and query document collections? One solution is a database. In the first part of this tutorial, we will discuss the problem of integrating XML with databases and examine choices, such as relational databases, object databases, object-relational databases and native XML servers. In the second part of this tutorial, we will discuss some of the issues in managing XML with Objects and present a generic architecture for storing XML in a relational database using detailed code examples in Java and Python.

### Attendee background

Attendees should have a good understanding of Objects, XML and Databases. Some previous experience using these technologies would also be helpful. The intended audience is professionals charged with the task of evaluating technologies.

### Format

This tutorial will be lecture based.

### Presenters

*Akmal B. Chaudhri has been working with Objects and Databases for over 10 years. He has been a regular presenter on Java, XML and Databases at a number of international conferences. He has edited the books “Object Databases in Practice”, Prentice-Hall, 1998, “Succeeding with Object Databases”, John Wiley & Sons, 2000, “Java and Databases”, Hermes Penton Science, 2002 and is currently working on a new book titled “XML Data Management”, Addison-Wesley, 2003. At present, he works for IBM developerWorks where he is also editor for the Open Source Projects web site. He has previously worked for Reuters, Logica, Computer Associates and Informix Software. He holds a B.Sc. in Computing & Information Systems, M.Sc. in Business Systems Analysis & Design and a Ph.D. in Computer Science.*

*Richard Edwards is a Research Officer at the School of Informatics at the University of Wales, Bangor in the UK. He graduated with a Ph.D. from Leeds University in 1995, following which he spent 4 years in industry as a software developer (specialising in relational databases, performance tuning, GUIs, OLAP & data warehousing) before returning to academia. His research interests include XML and databases, metadata, XML transformation using XSLT and the semantic mapping metalanguage Expressive / SML, and supporting online communities with groupware. He has also published and presented on XML Repositories.*

Tuesday 13:30-17:00

## 38 Distributed .NET

Michael Stal, *SIEMENS AG Corporate Technology*,  
*Michael.Stal@mchp.siemens.de*

Like the Java 2 Platform, the .NET Framework offers many types to support applications in many different domains. One of the most important areas where .NET Framework classes can be applied is distributed programming. For this purpose there is functionality such as .NET Remoting, XML Web services, ADO.NET, XML, COM interoperability, and much more. For the developer it is difficult to decide which technologies to use for which type of application as well as when and how to plug these technologies together.

In this tutorial the different parts of the .NET Framework for distributing and integrating components will be introduced in a systematic way. It will also be shown how these parts can be brought together to build distributed software systems.

### Attendee background

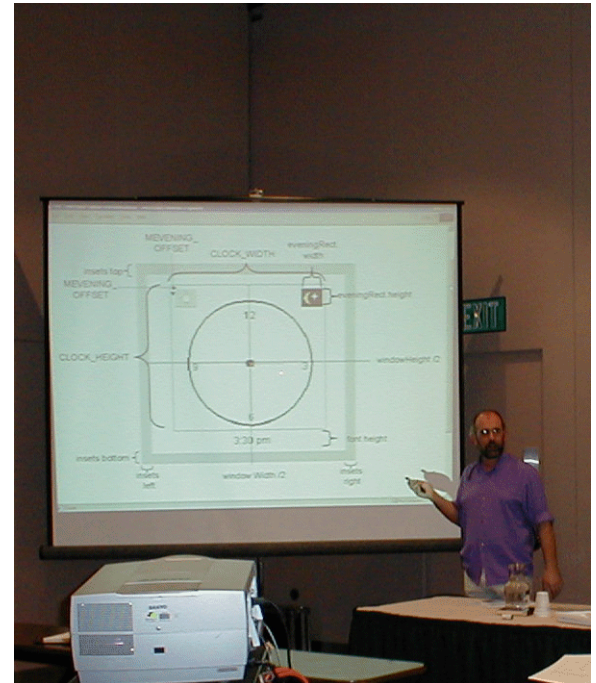
Basic knowledge in using Microsoft .NET. Language skills in C# or Java. Familiarity with distributed systems.

### Format

Slide presentation with small demos.

### Presenter

*Michael Stal is Senior Principal Engineer at Siemens responsible for research on Middleware and Application Integration. Michael focuses on Distributed Objects & Components, Software Architecture, and Web technologies. He is co-author of Pattern-Oriented Software Architecture Volume I&II, Siemens representative at the OMG, former member of the C++ X3J16 standardization working group, and editor-in-chief of Java Spektrum.*



## Wednesday, 6 November

10:00-17:30 Full Day

Wednesday 10:00-17:30

### 39 Ruby in a Day

Dave Thomas, *The Pragmatic Programmers, LLC*,  
dave@pragmaticprogrammer.com

Andy Hunt, *The Pragmatic Programmers, LLC*,  
andy@pragmaticprogrammer.com

Smalltalk was ahead of its time: we're just entering the decade of the untyped, flexible language. And by all accounts, Ruby could well be the language of that decade. Small, but tremendously expressive, Ruby is finding favor among all kinds of developers. From web applications to numerical simulations at NASA, Ruby is gaining popularity and mindshare.

As a developer, you owe it to yourself to have a look at Ruby. Even if you never write a line of Ruby code, the ideas in the language can greatly improve the way you think about design and the ways you implement your programs. And if you do start writing Ruby, you'll discover the tremendous productivity and readability gains that are possible.

#### Attendee background

Attendees will be familiar with the basics of object-oriented programming. We encourage those who can to bring laptops so we can jointly work on exercises and programming projects.

#### Format

Slides with a large number of exercises.

#### Presenters

*Dave Thomas has been developing software since the mid 1970's. He holds an honors degree in computer science from London University, and is a member of the IEEE Computer Society and the ACM. He ran a successful software company in the United Kingdom before moving to the United States and forming the Pragmatic Programmers with Hunt. Together they authored two books, "The Pragmatic Programmer: From Journeyman to Master" and "Programming Ruby: The Pragmatic Programmer's Guide". They are authors of the Manifesto for Agile Software Development, and jointly edit the Construction column for IEEE Software magazine. He is a keen private pilot.*

*Andy Hunt has been developing software since the early 1980's, in various senior positions at companies large and small, before becoming a consultant. He holds a BS in Information and Computer Science from the Georgia Institute of Technology and is a member of the IEEE Computer Society and the ACM. When not programming, he is an avid jazz musician and woodworker. Andy has authored two books with Dave Thomas: "The Pragmatic Programmer: From Journeyman to Master", and "Programming Ruby: The Pragmatic Programmer's Guide". They are authors of the Manifesto for Agile Software Development, and jointly edit the Construction column for IEEE Software magazine.*

Wednesday 10:00-17:30

### 40 Aspect-Oriented Programming with AspectJ

Erik Hilsdale, *PARC*, hilsdale@parc.com

AspectJ is a seamless aspect-oriented extension to Java. It can be used to cleanly modularize the crosscutting structure of concerns such as exception handling, multi-object protocols, synchronization, performance optimizations, and resource sharing.

When implemented in a non-aspect-oriented fashion, the code for these concerns typically becomes spread out across entire programs. AspectJ controls such code-tangling and makes the underlying concerns more apparent, making programs easier to develop and maintain.

This tutorial will introduce aspect-oriented programming and show how to use AspectJ to implement crosscutting concerns in a concise, modular way. We will use numerous examples to develop participant's understanding of aspect-oriented programming through AspectJ, and hands-on exercises to enhance that understanding. We will also demonstrate and use AspectJ's integration with IDEs such as JBuilder 6 and Forte4J, and emacs, in addition to the core AspectJ tools.

AspectJ is freely available at <http://aspectj.org>.

#### Attendee background

Attendees should have experience doing object-oriented design and implementation, and should be able to read Java code. No prior experience with aspect-oriented programming or AspectJ is required.

#### Format

This tutorial will have both a lecture component and a hands-on programming component.

#### Presenter

*Erik Hilsdale is a researcher at the Palo Alto Research Center. As a member of the AspectJ team, he focuses on language design, pedagogy and compiler implementation. He has written several conference and workshop publications in programming languages. He is an experienced and energetic instructor in programming languages with a long background with AspectJ.*



**Wednesday 10:00-17:30**

---

**41 Programming Web Services for Mobile Devices**

---

Scott Guthrie, *Microsoft*, danf@microsoft.comDan Fay, *Microsoft Research*, danf@microsoft.comSimon Cuce, *Monash University*, simon.cuce@csse.monash.edu.au

The Internet is quickly evolving from today's websites that just deliver UI pages to browsers to the next generation of programmable websites that directly link organizations, applications, services, and devices with one another. These programmable websites become more than passively accessed sites - they become reusable, intelligent web services. With the beginning of the XML Web Services revolution, the landscape of the internet is changing, and understanding how to make information available through web services to any type of mobile device is paramount to widespread end-user adoption.

This tutorial will begin by covering the basics of web services and all the relevant standards (SOAP, WSDL, and UDDI) associated with creating and deploying a web service. We will demonstrate to attendees how to build and deploy web services and client applications with an XML Web Service that uses ASP.NET and standard internet technologies.

Attendees will also discover how to take advantage of web services from mobile devices to create both rich web applications via the Mobile Internet Toolkit as well as compelling client applications via the .NET Compact Framework.

**Attendee background**

Participants should have a general familiarity with web technologies and object-oriented programming.

**Format**

This tutorial will be lecture based.

**Presenters**

*Scott Guthrie co-founded the Microsoft ASP.NET team which is responsible for web services and leads the design team responsible for architecting the product. His individual technical contributions include: ASP.NET Web Services Infrastructure, ASP.NET Web Forms Page Architecture, ASP.NET Compilation System, and the ASP.NET HTTP Runtime Architecture.*

*Dan Fay is a 10 year veteran of Microsoft working with researchers looking at new internet technologies. Dan also supports academic research and development on the .NET Framework, including the Common Language Runtime and Web Services.*

*Simon Cuce is a final year Ph.D. student for the Dept. of Computer Science and Software Engineering at Monash University, Australia. He is also a Senior Consultant for ProAzial Solutions Pty. Ltd., one of Australia's premiere .NET software consultants and handheld application developers. He has a background in mobile technology, including handheld and mobility enabling software development. Simon also came in third place in the 10th ACM International Graduate Student Research Contest 2002 for his research project GLOMAR: Adaptive Consistency Control for Distributed File Systems.*



## 10:00-13:15 Morning

Wednesday 10:00-13:15

**42 Refactoring To Patterns**Joshua Kerievsky, *Industrial Logic, Inc.*, [joshua@industriallogic.com](mailto:joshua@industriallogic.com)

Patterns are undeniably useful design aids and a cornerstone of object-oriented programming. Yet with the emergence of agile methodologies like Extreme Programming, our motivation for using patterns has changed. Today, we use patterns when they can help us reduce or remove duplication, simplify the un-simple or make our code better at communicating its intents. In other words, we refactor to patterns for the same reason we refactor in general: to improve the design of our code. But since the motivation for using a pattern, according to the published literature, rarely resembles the motivation to refactor, it is easy to misapply patterns or over-engineer solutions using patterns. In this tutorial, we will study numerous real-world refactorings to patterns (in Java and UML) and examine the refactoring roots of patterns, the role of patterns in evolutionary design and common motivations for refactoring to patterns.

**Attendee background**

Attendees will get the most out of this tutorial if they have some basic experience with Refactoring and Design Patterns.

**Format**

Interactive lecture.

**Presenter**

Joshua Kerievsky is a software development coach and programmer. After programming on Wall Street for nearly 10 years, in 1995 he founded San Francisco Bay Area-based Industrial Logic (<http://industriallogic.com>), a company that specializes in Extreme Programming (XP). Since 1999, Joshua has been programming and coaching on small, large and distributed XP projects and teaching XP to people throughout the world. He regularly speaks about XP, has authored numerous articles, simulations and games about XP and patterns and is working on the forthcoming book *Refactoring to Patterns* (<http://industriallogic.com/xp/refactoring/>).

Wednesday 10:00-13:15

**43 EJB Roleplay**Nicolai Josuttis, *Solutions in Time*, [nico@josuttis.com](mailto:nico@josuttis.com)Jutta Eckstein, *Objects in Action*, [jeckstein@acm.org](mailto:jeckstein@acm.org)

Do you know Enterprise Java Beans? Do you know about how containers manage the lifecycle of EJBs? Do you understand the difference between entity beans and stateless session beans? Do you know why we didn't mention stateful session beans in the previous question? EJB roleplay is a way to understand the lifecycle of EJBs. By interacting with others you "feel" the way things are organized. It's an interesting way to learn complex behavior for both newbies and experts.

**Attendee background**

Attendees should have some basic knowledge of the following EJB concepts: the difference between Entity and Session Beans, Remote Interface, Bean class, and Primary Key. In addition, you shouldn't be nervous about being active in the tutorial or about participating in role-playing learning in front of others.

**Format**

Role-playing (no lecture).

**Presenters**

Nicolai Josuttis is an independent systems architect, author, and consultant. He designs mid-sized and large software systems for the telecommunication, traffic, finance, and manufacturing industries. He is well known for his best-selling book about the C++ Standard Library. But he has also written other books and articles about object-oriented software development and programming in general. He is a partner at System Bauhaus, a German group of recognized object-oriented system development experts.

Jutta Eckstein is an independent consultant and trainer from Munich, Germany. She has over ten years experience in developing object-oriented applications. In addition to engineering software she has been designing and teaching OT courses in industry. Having completed a course of teacher training and having led many 'train the trainer' programs in industry, she focuses also on techniques for teaching OT, and is a lead in the pedagogical patterns project. She has presented her work at OOPSLA, OT, and EuroPloP.

Wednesday 10:00-13:15

**44 Test-Driven Development in .NET**James Newkirk, *Thoughtworks, Inc.*, [newkirk\\_james@hotmail.com](mailto:newkirk_james@hotmail.com)

This tutorial presents techniques for writing software in Microsoft .NET using the extreme programming principles of test-driven development, and refactoring. It will be conducted using real world examples to illustrate key techniques. These techniques can be successfully applied in the following areas of interest in the .NET environment: Web Services, and ASP.NET. In order to demonstrate how to do test-first in these areas the open source NUnit testing framework (<http://nunit.sourceforge.net>) will be described and used in the tutorial.

**Attendee background**

Participants should have a general understanding of the XP coding principles of Test-First Design, and Refactoring. Familiarity with the C#, and ASP.NET would be useful, but not required.

**Format**

This tutorial will be lecture based.

**Presenter**

James Newkirk is a Software Development Manager/Architect with eighteen years development experience. His industry experience includes real-time micro-controllers to web services. He has published in both language and management publications and spoken at many industry conferences, including a previous OOPSLA conference. He also co-wrote *Extreme Programming in Practice*, published by Addison-Wesley, 2001. Since August of 2000 he has been working with the .NET Framework and has contributed to the development of NUnit a unit-testing tool for .NET.

13:30-17:00 Afternoon

Wednesday 13:30-17:00

## 45 Object-Oriented Reengineering: Patterns & Techniques

Serge Demeyer, *University of Antwerp*, [serge.demeyer@ua.ac.be](mailto:serge.demeyer@ua.ac.be)  
 Stéphane Ducasse, *University of Berne*, [ducasse@iam.unibe.ch](mailto:ducasse@iam.unibe.ch)

Surprising as it may seem, many of the early adopters of the object-oriented paradigm already face a number of problems typically encountered in large-scale legacy systems. Software engineers are now confronted with millions of lines of industrial source code, developed using object-oriented design methods and languages of the late 80s and early 90s. These systems exhibit a range of problems, effectively preventing them from satisfying the evolving requirements imposed by their customers.

This tutorial will share our knowledge concerning the reengineering of object-oriented legacy systems. We will show you techniques and tools we have applied on real industrial OO systems to detect and repair problems. In particular, we will discuss issues like reverse engineering, design extraction, metrics, refactoring and program visualisation.

### Attendee background

Participants should have practical programming experience in at least one OO language (Smalltalk, C++, Java, Eiffel, ...). Familiarity with UML is useful, though not required.

### Format

This tutorial will be lecture based, but involves a large amount of interaction with the audience.

### Presenters

*The presenters have presented tutorials on Object-Oriented Reengineering at both OOPSLA and ECOOP and have written a book entitled "Object-oriented Reengineering Patterns" (Morgan Kaufmann Publishers, 2002).*

*Serge Demeyer is a professor in the Department of Mathematics and Computer Science at the University of Antwerp in Belgium. There he leads a research group investigating the theme of "Software Reengineering" (LORE - Lab On REengineering). His main research interest concerns software engineering (more precisely, reengineering in an object-oriented context) but due to historical reasons he maintains a heavy interest in hypermedia systems as well. He is an active member of the corresponding international research communities, serving in various conference organization and program committees.*

*Stéphane Ducasse is a post-doctoral researcher in the Software Composition Group in Berne. He served as technical leader of the FAMOOS Esprit project; a project whose goal it was to come up with a set of reengineering techniques and tools to support the development of object-oriented frameworks. He is an expert in object-oriented programming, design patterns, framework development, reflective programming and component technology. He is one of the main designers of the MOOSE reengineering environment that is the basis for CodeCrawler, a program understanding tool. He is the main organizer of the annual European Smalltalk Advanced Seminars.*

Wednesday 13:30-17:00

## 46 Patterns for EJB Development

Bobby Woolf, *Cyberdyne Software, Inc.*, [woolf@acm.org](mailto:woolf@acm.org)

This tutorial discusses eight patterns that will help participants understand how to better design their Enterprise JavaBean (EJB) code. It focuses on the services layer, the most critical part of a multi-tier application architecture because it defines the client's API to the server. The patterns document simple but powerful techniques for designing code that is more reusable, more consistent, and easier to maintain. The tutorial includes detailed code examples to illustrate concepts. Once the patterns are explained, the tutorial shows a cookbook for how to use all of the patterns together. The coding techniques are completely EJB standard compliant and produce code that will deploy into any J2EE/EJB container (WebLogic, WebSphere, JBoss, etc.).

### Attendee background

Participants should be experienced Java programmers and beginner-to-intermediate EJB developers, or technical leads that need to mentor such developers.

### Format

This tutorial will be lecture based and include code examples.

### Presenter

*Bobby Woolf, an independent consultant, mentors teams developing object-oriented, distributed, multi-tier applications using Java/J2EE and related technologies. His interests include patterns and agile methodologies.*

Wednesday 13:30-17:00

**47 Scrum and Agile Process 101**Ken Schwaber, *ADM*, [ken.schwaber@verizon.net](mailto:ken.schwaber@verizon.net)

Agile processes are different. They not only increase productivity, they bring focus and pleasure back to systems development. This tutorial explains the underlying theory and practices of all agile processes, and then explains how they are implemented in Scrum. A case study is presented. An exercise is then conducted to give tutorial attendees a feel for the flow, practices, and rules of Scrum. Tutorial attendees are organized into teams that collaborate with the customer to formulate an iteration, self-organize to identify the work in the iteration, report on progress and impediments during the iteration, and present the results of the iteration to the users.

**Attendee background**

Previous participation in a software development project.

**Format**

Lecture and exercises.

**Presenter**

*Ken Schwaber is one of the developers of the Scrum agile process and has extensively used agile processes over the last seven years. Ken is one of the founders of the AgileAlliance and helped setup the AgileAlliance organization. In 2001, Ken co-authored "Agile Software Development with Scrum" with Mike Beedle. With over thirty years of software development, management and consulting experience, Ken is currently working with organizations to develop software with Scrum and a combination of Scrum and Extreme Programming, as well as helping the organizations plan and execute the required change management.*

Wednesday 13:30-17:00

**48 Use Cases and Testing: Using Use Cases to Write Test Cases**Jim Heumann, *Rational Software*, [jheumann@rational.com](mailto:jheumann@rational.com)

Use cases, a proven technique for specifying functional requirements, are particularly powerful because of their ability to be easily used by other members of the software development team. This is true in testing, where requirements form the basis for determining whether system objectives and quality measures have been met. The ability to create test cases from, and trace them to, use cases is a vital skill for ensuring a quality product. This tutorial will introduce a straightforward technique, based on the Rational Unified Process, for generating test cases directly from use cases. The tutorial will start with a brief introduction to use cases including guidelines for content and format to facilitate testing, and will continue by explaining the three key steps used for generating test cases: finding use case scenarios, identifying conditions and data elements to test, and adding data values for testing.

**Attendee background**

A familiarity with use cases would be helpful, but not required.

**Format**

Lecture/presentation interspersed with discussion of the examples and an in-class exercise.

**Presenter**

*Jim Heumann is Requirements Management Evangelist for Rational Software. Throughout his twenty year career as a designer, implementer, maintainer, consultant, and manager he has been able to experience, first-hand, most of the challenges we all face day-to-day as software professionals. He has worked on projects in artificial intelligence, cell phone fraud detection, user interface design, visual data analysis, currency trading, image processing, and wind profilers, among other. Jim has delivered over fifty workshops, tutorials, training classes, and conference presentations over the last ten years on a variety of subjects including Requirements Management with Use Cases, Object-Oriented Analysis and Design, Visual Data Analysis, UML, and Use Cases and Test Cases. He holds an MS in Management Information Systems from the University of Arizona.*

## Thursday, 7 November

13:30-17:00 Afternoon

Thursday 13:30-17:00

### 49 How to Use Design Patterns In Java and .NET

James Cooper, *IBM T. J. Watson Research Center*,  
jwcnmr@watson.ibm.com

Design patterns are a part of the toolbox of the practical application programmer. While at first, some programmers felt them more abstractions than tools, patterns are in fact ideal for writing good object-oriented programs in highly object-oriented languages such as Java, C# and VB.Net. In fact, these approaches provide a way for classes to communicate without becoming entangled. This tutorial will outline the OO approaches you can use in all 3 languages, thus bridging the gap between Java and C# programming, and will indicate some of the major language similarities and differences. We will then illustrate a number of common design patterns by developing several easy to grasp visual programs in Java and .NET. We will provide simple examples of at least the Command, Mediator, Iterator, Adapter, Bridge, Observer, Template, Factory method and Abstract Factory patterns. We'll also show how to use the Façade pattern to simplify database access. In each case, we will build a working, visual program in at least 2 of the 3 languages, and discuss its advantages and pitfalls. Examples of all patterns in all 3 languages will be available on a CD-ROM.

#### Attendee background

Attendees should be familiar with programming at an intermediate level in Java, C# or VB.Net, and should be comfortable with basic concepts of classes and objects, interfaces and inheritance. They do not need to have any prior knowledge of Design Patterns.

#### Format

Lecture with examples of running programs. Ample time for question and answers will be provided and if we can make CDs of code available to attendees in advance, they are welcome to run the samples on their own systems and ask questions as time permits.

#### Presenter

*James Cooper is a researcher at IBM T J Watson Research, in unstructured information management and retrieval. He is the author of 15 books, including Java Design Patterns: A Tutorial, Visual Basic Design Patterns (VB6 and VB.Net) and Design Patterns in C#, all from Addison-Wesley. He is also a columnist for JavaPro magazine, where he frequently discusses design patterns. He has presented over 30 lectures and tutorials on Design Patterns in Java at JavaONE, JavaEdge, SIGS Java conferences and the Java Software Summit, and at various colleges. He has published extensively in the field of information retrieval, and holds a number of patents.*

Thursday 13:30-17:00

### 50 Daily Builds are for Wimps

Michael Two, *Thoughtworks Inc.*, 2@thoughtworks.com

Over the past few years we have been working on a very large J2EE application with over 25 developers and several hundred thousand lines of code. Our original build process took a few hours and was very fragile. This caused us to go without a working build for days at a time. We successfully shrank the time to well under an hour and we produce more than 20 builds a day with failures being very rare. Our fully automated process allows developers to check in code anytime and find out the results of compilation, unit testing, deployment and automated acceptance testing of the build as soon as it is available. All of the results are available on a web page that all members of the project can use to monitor the state of the application. We will cover building your project in ANT, extending ANT by writing your own custom tasks and the open source tool Cruise Control for managing your build. Examples and demonstrations will be provided in Java and .NET for the building and testing of a simple web service using ANT, JUnit, NANT and NUnit.

#### Attendee background

Attendees should have a working knowledge of Java or C#. Familiarity with XML syntax and web services is also helpful.

#### Format

Lecture

#### Presenter

*Michael has been a developer at Thoughtworks for three years. He has worked on large scale J2EE projects with teams of up to 60 people. He has been lecturing on Continuous Integration since August 2000 at places like XP Immersion IV, SDWest 2001 and 2002, SDEast 2001, OOPSLA 2001 and NASA.*

Thursday 13:30-17:00

**51 Successful Technical Leading on an OO project**Andrew Schneider, BJSS Ltd., [as@bjss.co.uk](mailto:as@bjss.co.uk)

So you're a technical lead. You will have to combine many disparate skills to achieve that elusive goal, the successful team. Project management, people management, requirements management, as well as good design and implementation skills are all core ingredients. If you have ever pondered any of the following questions this tutorial is for you:

- How much design is enough?
- How much project planning is enough?
- How do I select and introduce appropriate technologies and methods?
- What sort of leadership style is appropriate to my team?
- How do I gain the respect I need to lead the team?
- How much freedom should the team have?
- How do I handle problem team members?
- How do I handle a project in crisis?
- How do I handle the age-old "I want it fast, flexible, cheap and now" problem?

This tutorial will answer these questions and more. It will provide a holistic view of technical leadership. During the tutorial we'll outline tools and techniques such as flexibility matrices, worry-first management (a risk-focused approach to management) and targeted design. Participants can also share experiences and will leave with a comprehensive set of references for further exploration.

**Attendee background**

Participants will be either new to technical leadership or already have been in a technical lead role for some time and wish to share experiences and learn some new techniques.

**Format**

Lecture with group discussion and copious real world examples.

**Presenter**

*Andy Schneider wears consultant and architect hats for BJSS, a software services organisation. He has been involved in implementing, designing and leading projects utilising object orientated technology since 1990. His primary interests are complex distributed systems, security and technical management. When wearing his consultant hat he finds himself performing presentations and tutorials on a regular basis.*

Thursday 13:30-17:00

**52 Notes on the forgotten Art of Software Architecture**Frank Buschmann, Siemens AG, Corporate Technology,  
[Frank.Buschmann@mchp.siemens.de](mailto:Frank.Buschmann@mchp.siemens.de)

Everybody is praising the benefits of contemporary distributed object computing and component middleware. CORBA, J2EE, .NET, and other platforms are used in everybody's working environment. All these platforms promise to make software development easy, productive, and successful. Unfortunately, it is precisely these benefits that are these platforms' biggest drawbacks! For example, it is true that CORBA, J2EE, and .NET make it easy to access a remote component and use its services. This apparent "easyness," however, raises the illusion that it is a child's game to master the inherent complexities of large-scale distributed systems with stringent end-to-end quality of service requests, just by passing the responsibilities for these concerns to the middleware. The consequence of such incorrect beliefs is that managers, designers, and developers tend to pay less and less attention to the architecture of the systems they are building. Ultimately, this leads to the failure of these projects.

All contemporary platforms can only help you in doing your job. They will not do your job for you. Software architectures must be crafted with care!

This tutorial therefore explores some of the secrets of building high-quality software architectures, in terms of methodology, design goals, and architectural properties, in order to recall the foundations of the forgotten art of building successful software.

**Attendee background**

Development and design experience.

**Format**

Slides, tutorial style but with much interaction with the audience.

**Presenter**

*Frank Buschmann is senior principal engineer at Siemens Corporate Technology in Munich, Germany. His interests include Object Technology, Frameworks and Patterns. Frank has been involved in many software development projects. He is leading Siemens' pattern research activities. Frank is co-author of "Pattern-Oriented Software Architecture -- A System of Patterns" and "Pattern-Oriented Software Architecture -- Patterns for Concurrent and Networked Objects"*



Thursday 13:30-17:00

## 53 Framework Design and Implementation using Java and UML

Dirk Riehle, *SKYVA International*, [dirk@riehle.org](mailto:dirk@riehle.org)

Alan Perry, *SKYVA International*, [aperry@skyva.com](mailto:aperry@skyva.com)

This tutorial shows how to design and implement object-oriented frameworks using a real-world example, Java, and UML. The example is the open-source framework, JValue, which we will see evolve over several steps. Topics discussed and explained are: effective interface design and abstract superclasses for code reuse, framework types, interface and implementation architecture of frameworks, design patterns and frameworks, framework extension and framework layering, and framework evolution under benign circumstances (you have control over clients). Specific emphasis is put on the use of UML collaborations to model design patterns and frameworks, giving us a much clearer picture of how to separate a framework from its use-context.

### Attendee background

Attendees are expected to be familiar with Java and UML on an intermediate level.

### Format

Lecture style.

### Presenters

*Dirk Riehle is a software developer at SKYVA International, where he is developing business software based on object-oriented frameworks. Dirk has worked on design patterns, object-oriented frameworks, and software architecture for the last 10 years. More recently, he became the main designer and implementer of the first UML Virtual Machine using a framework approach. At OOPSLA, he may be best known for his technical paper presentations on the subject matter. Dirk holds a Ph.D. from Swiss Federal Institute of Technology and welcomes feedback through [dirk@riehle.org](mailto:dirk@riehle.org).*

*Alan Perry is a senior developer at SKYVA International. After studying physics he chose a career in software, and has worked in the fields of semiconductor CAD, Laboratory Information Management Systems, and now in the design and production of business applications. He has worked with various frameworks for over 5 years and has been one of the main outside influences driving the evolution of the JValue framework used as the main example in this tutorial.*

Thursday 13:30-17:00

## 54 Developing Web Services

Qusay Mahmoud, *Simon Fraser University, Vancouver, Canada*,  
[qmahmoud@cs.sfu.ca](mailto:qmahmoud@cs.sfu.ca)

Web services are services offered through the Web and can be accessed from any web service-enabled machine with Internet access. Web services enable interoperability through a set of XML-based open standards. Businesses use the XML-based Web Services Description Language (WSDL) to describe their web services on the Internet and list them in an XML-based registry such as the Universal Description, Discovery, and Integration (UDDI) protocol. UDDI allows you to find publicly available web services. A client sends a service request to the directory, which in turn informs the client about the services available. The SOAP protocol is then used to communicate, using HTTP and XML as an exchange mechanism, between the applications running on different platforms. This tutorial will help participants understand web services and their usefulness. A service-based architecture will be discussed. The tutorial will also help participants get a feeling of the effort involved in developing web services.

The list of major topics to be covered in this tutorial includes:

- SOAP
- UDDI
- WSDL
- Security issues
- Deploying web services on J2EE

See

<http://developer.java.sun.com/developer/technicalArticles/webservices/ws2ee/>.

### Attendee background

Knowledge of the Java programming language, as examples of web services will be shown using Java.

### Format

Presentation based (but will show live examples).

### Presenter

*Qusay H. Mahmoud is a faculty member in the School of Computing Science at Simon Fraser University, Canada. He has published dozens of articles on the Java programming language, including wireless Java articles for Sun Microsystems, and articles on deploying web services on J2EE. He is the author of 'Distributed Programming with Java' (Manning Publications, 1999), and 'Learning Wireless Java' (O'Reilly, 2002). Qusay has presented tutorials on wireless software development at several major international conferences worldwide.*

# WORKSHOPS

**Chair: Jutta Eckstein, Objects in Action,**  
[workshops@oopsla.acm.org](mailto:workshops@oopsla.acm.org)



The workshop program for OOPSLA 2002 includes a large number of workshops that spans a wide range of relevant and timely topics. Workshops provide intensive collaborative environments where object technologists meet to surface, discuss, and solve challenging problems facing the field. The topics covered by workshops are diverse, as are the workshop's formats. A workshop may provide the opportunity for representatives of a technical community to coordinate efforts and to establish collective plans of action, to collectively work on a book, or to discuss and share ideas on a hot new emerging technology.

To ensure a sufficiently small group for effective interaction, workshop organizers manage attendance based on an objective criterion, typically a short position paper submitted by potential attendees. In the position paper the prospective attendees outline their opinions on an aspect of the workshop's topic. Participants have been chosen based on the relevance of their position paper to the workshop scheme. Workshop attendance is by invitation only and it is at the discretion of the organizers but all attendees are expected to contribute to the discussion. After the workshop, the organizers are responsible for reporting results to the object community via a short summary that will appear on the workshop's web site.

Workshops are either full or half day events that occur on the first two days before the conference. Workshop organizers and participants must check-in at the registration desk before attending the workshop, preferably the evening before. For more information on a particular workshop, please visit its web page.

## Monday, 4 November

### 8:30-17:00

- 1 Tool Support for Aspect Oriented Software Development, *page 78*
- 2 2nd OOPSLA Workshop on Domain-Specific Visual Languages, *page 78*
- 3 Commonalities of Agile Methodologies, *page 79*
- 7 Top 10 issues of an open software model for embedded systems, *page 79*
- 8 The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer, *page 79*
- 9 Pervasive Computing; going beyond Internet for small screens, *page 80*
- 10 Distributed eXtreme Programming, *page 81*
- 12 Patterns for Software Architecture, *page 81*
- 14 Agent-oriented methodologies, *page 81*

## Monday (cont.)

- 15 Tackling the Discovery Costs of Evolving Software Systems, *page 81*
- 22 J2EE vs .NET, *page 82*
- 23 Using Play to Enhance Learning About Objects, *page 82*
- 24 1st International Workshop on Runtime Kernel Support for Dynamic Languages and Component Based Architectures, *page 82*
- 25 "Killer Examples" for Design Patterns and Objects First, *page 82*

### 8:30-12:00

- 5 Extreme Programming Practices in the First CS Courses, *page 83*

### 13:30-17:00

- 31 Expanding the Boundaries of Unit Testing, *page 83*

## Tuesday, 5 November

### 8:30-17:00

- 4 Generative Techniques in the context of Model-Driven Architecture, *page 84*
- 6 Architecture is dead - Long live the Architect!, *page 84*
- 11 Patterns in Distributed Real-Time and Embedded Systems, *page 84*
- 13 Patterns for Customer Interaction and Expectation Management, *page 85*
- 16 1st Workshop on Software Development Process Patterns, *page 85*
- 17 Built for Life: Constructing Software to Outlive Its Creators, *page 85*
- 18 Semantics of Enterprise Integration II, *page 86*
- 19 Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE), *page 86*
- 20 Object-Oriented Web Services, *page 87*
- 21 Software Apprenticeship: the Journey to Mastery, *page 87*
- 26 Using Domain Specific Languages to Drive Business Applications, *page 87*
- 27 Extravagaria - Art Assisting Science, *page 87*
- 28 Agile Processes Workshop II - Managing Multiple Agile Projects, *page 88*
- 30 Product Line Engineering - The early steps: Planning, Managing, and Modeling (PLEES'02), *page 88*

### 8:30-12:00

- 29 Pair Programming Explored, *page 88*

## Monday, 4 November

8:30-17:00 Full day

Monday 8:30-17:00

### / Tool Support for Aspect Oriented Software Development

Mark Chu-Carroll, *IBM T. J. Watson Research Center*,  
mcc@watson.ibm.com

Gail Murphy, *University of British Columbia, Department of Computer Science*,  
murphy@cs.ubc.ca

Aspect-oriented software development (AOSD) is a new approach for managing complexity in large software systems. Aspect-oriented techniques extend techniques such as object-orientation, enabling a software developer to modularize concerns (aspects) that crosscut the system's structure. Aspects may be described and manipulated at multiple points in the development process, from requirements to design to implementation.

To date, AOSD forums have largely focused on the design of languages and composition tools to allow developers to use aspect-oriented ideas in their code. To make aspect-oriented development an industrial-strength technology, the concepts and techniques of aspect-orientation need to be integrated into tools used throughout the software process --- including design and modeling tools and techniques, software configuration management systems, programming environments, debuggers, and deployment tools.

In this workshop, we will discuss what tools are needed to support aspect-oriented development, and how that support can best be integrated into tools in a consistent way.

Workshop URL:  
<http://www.cs.ubc.ca/~murphy/OOPSLA02-Tools-for-AOSD/>.

Monday 8:30-17:00

### 2 2nd OOPSLA Workshop on Domain-Specific Visual Languages

Juha-Pekka Tolvanen, *MetaCase Consulting*, jpt@metacase.com

Jeff Gray, *University of Alabama at Birmingham*, gray@cis.uab.edu

Matti Rossi, *Helsinki School of Economics*, mrossi@hkkk.fi

An upward shift in abstraction leads to a corresponding increase in productivity. In the past this has occurred when programming languages have evolved towards a higher level of abstraction. Today, domain-specific visual languages provide a viable solution for continuing to raise the level of abstraction beyond coding, making development faster and easier.

In a domain-specific visual language, the models are made up of elements representing concepts that are part of the domain world, not the code world. The language follows the domain abstractions and semantics, allowing developers to perceive themselves as working directly with domain concepts.

Workshop topics:

- Industry/academic experience reports
- Approaches to identify constructs for domain-specific languages
- Novel approaches for code generation from domain-specific models
- Issues of support/maintenance for systems built with DSL's
- Evolution of languages in accordance with domain
- Meta-modeling frameworks and languages
- Tools for supporting domain-specific languages

Workshop URL: <http://www.cis.uab.edu/info/OOPSLA-DSVL2/>.

Monday 8:30-17:00

### 3 Commonalities of Agile Methodologies

Jens Coldewey, *Coldewey Consulting*, [jens\\_coldewey@acm.org](mailto:jens_coldewey@acm.org)  
 Pete McBreen, *McBreen Consulting*, [petemcbreen@acm.org](mailto:petemcbreen@acm.org)  
 Klaus Marquardt, *Dräger Medical AG*, [marquardt@acm.org](mailto:marquardt@acm.org)

With the increasing interest in agile development the various methodologies have gained more and more attention in the community. As a side effect of their success, nearly every process around tries to prove that it is agile, whatever that may be. Although the agile manifesto defines the core values of agile development, it is easy to state that a particular process values humans - and an insult to argue against that.

With all their differences, published agile methodologies, such as ASD, Crystal, Scrum, or XP, have several techniques in common, other processes don't have. One example is a meeting structure with different meetings that reflects the value agile development assigns to collaboration. The objective of this workshop is to identify how the principles of agile development are reflected in practice in Agile projects - both to help evaluating existing methodologies and to help setting up a project-specific process.

Workshop URL:

<http://www.coldewey.com/publikationen/conferences/oopsla2002/CommonalitiesWorkshop.htm>.

Monday 8:30-17:00

### 7 Top 10 issues of an open software model for embedded systems

Peter Kriens, *Aquite*, [peter.kriens@aquite.se](mailto:peter.kriens@aquite.se)  
 Niclas Nilsson, *Activa AB*, [niclas.nilsson@activa.se](mailto:niclas.nilsson@activa.se)  
 Rob van den Berg, *VDO*, [Rob.vandenBerg@NL2.vdogrp.de](mailto:Rob.vandenBerg@NL2.vdogrp.de)  
 BJ Hargrave, *IBM Pervasive Computing*, [hargrave@us.ibm.com](mailto:hargrave@us.ibm.com)

Embedded devices are becoming so powerful and connected that a popular "third party" application model will develop in the future. Such a model allows developers to target a large range of embedded devices instead of the current proprietary model. This workshop wants to investigate the unique problems associated with such a model and investigate the efforts that are under way. Participants of this workshop should play a role in this developing field or have a strong interest in the outcome:

- Carlets, Xlets, Midlets
- Developers involved Windows CE, Microsoft Embedded XP
- General Agent frameworks
- General Component frameworks
- Patterns for embedded systems
- OSGi
- Large scale management and deployment

The workshop will produce a list of 10 top problems to be resolved for this area to mature, and will provide a possible roadmap to resolve those problems.

Workshop URL: <http://www.carbundles.com>.

Monday 8:30-17:00

### 8 The 11th OOPSLA Workshop on behavioral semantics -- Serving the customer

Haim Kilov, *Financial Systems Architects*, [haimk@acm.org](mailto:haimk@acm.org)  
 Kenneth Baclawski, *Northeastern University*, [ken@baclawski.com](mailto:ken@baclawski.com)

The goal of this workshop is to foster precise and explicit OO specifications of business and system semantics, independently of any (possible) realization. Substantial progress has been made in these areas, both in academia and in industry. However, in too many cases only lip service to these ideas has been provided, and as a result the systems we build or buy are all too often not what they are supposed to be. Doing better than that requires both a clear understanding of the semantics of the problems together with their business and technological environments, and abstract, precise and explicit specifications of that semantics. The specific theme this year is on serving the customer. As in our previous workshops, we want to bring together theoreticians and practitioners to report their experience with making semantics precise, clear, concise and explicit in OO business specifications, business designs, and system specifications. Papers can range from academic research to industrial "war stories."

Workshop URL: <http://www.ccs.neu.edu/home/kenb/oopsla2002>.

**Monday 8:30-17:00**

## 9 Pervasive Computing; going beyond Internet for small screens

Steffen Schaefer, *IBM UBG*, [steffens@acm.org](mailto:steffens@acm.org)

Steve Marney, *EDS Bluesphere*, [steve.marney@eds.com](mailto:steve.marney@eds.com)

Johan Vos, *Acunia*, [johan@acunia.com](mailto:johan@acunia.com)

Pervasive Computing has become a reality. Over the past months numerous wireless portals and wireless applications have been built and deployed for a growing number of users. Almost every week, new device types and services become available. And even though Pervasive Computing is still in its infancy, no one can deny its potential to change how we live and do business.

Pervasive Computing and the supporting technology available today all point toward a future, in which the networking web will be extended to an ever larger number of players, extending to smaller machines, appliances, etc., which are increasingly becoming more intelligent. This web will allow machines to interact with other machines and humans. The sheer number of players interacting with each other and their penetration in our daily lives, e.g., microwave, washing machine, light bulb, is going to make the impact even more formidable.

In some respects, pervasive computing solutions are not very different from standard e-Business systems and are implemented with the same or similar technology. However, the combination of the pervasiveness of computing functionality and Internet access, together with the possibility to monitor events and push information to users, and the possibility to create location-aware services, is a new paradigm.

We believe Object Technology is playing a major part in shaping up this new computing paradigm. As in other e-Business systems, Java, C++ and XML are often used for implementation at the server side, and in the meantime as well as on the pervasive device.

In this workshop, we will identify recurring themes and patterns, we will categorize some typical solutions observed and discuss their reference architectures. Clearly, the standard Web 'Request/Response' model, e.g., as implemented through WAP is only one way of providing pervasive applications. Offline-work, Messaging, and Business-To-Machine are just a few examples where Pervasive Computing goes beyond 'Internet for small screens'.

The workshop will bring together practitioners who have been actively involved in the development of Pervasive Computing solutions, researchers who have been working in this area, and people who have been involved in the definition of standards. Broadly speaking, the goal is to discuss experiences and findings, share ideas and build a community in which we can continue discussion after the workshop. We will aim to produce tangible results, e.g., in terms of architecture or design patterns for specific issues, and try to reach consensus on approaches for resolving problems and identify strategies.

Topics of our discussions might include, but are not limited to:

- Application architecture for client devices and back-end service provider platforms
- Wireless Portals
- Mobile Applications, e.g. using WAP, SIM Toolkit, i-Mode, SMS,...
- Applications for PDAs, Handhelds, Smartphones, Cars, and other devices
- Specific mobile related topics such as Content Adaptation / Transcoding, Location Based Services, Content Management or Personalization for mobile devices
- Mobile Agents
- (Micro)-Payments, mobile e-Commerce
- Management and administration of software deployed on a large number of heterogeneous devices
- Security and Privacy
- Supporting dramatically different user interface paradigms
- Adaptability and learning
- Networking infrastructure

Workshop URL: <http://www.jeckstein.com/oopsla/pervasive-computing>.



Monday 8:30-17:00

**10 Distributed eXtreme Programming**

Prashant Jain, *Manystreams, Inc.*, [prashant@noida.manystreams.com](mailto:prashant@noida.manystreams.com)  
 Michael Kircher, *Siemens AG, Corporate Technology*,  
[Michael.Kircher@mchp.siemens.de](mailto:Michael.Kircher@mchp.siemens.de)  
 David Stotts, *University of North Carolina at Chapel Hill*,  
[stotts@cs.unc.edu](mailto:stotts@cs.unc.edu)  
 Laurie Williams, *North Carolina State University*,  
[lawilli3@unity.ncsu.edu](mailto:lawilli3@unity.ncsu.edu)  
 Alan Wills, *FastNLoose*, [alan@fastnloose.com](mailto:alan@fastnloose.com)

Distributed eXtreme Programming (DXP) is eXtreme Programming with a relaxation on the requirements of close physical proximity of the team members. DXP is a fairly new concept that is increasingly being used across many companies. The aim of this workshop is to bring together practitioners who have tried using DXP as well as those who have used XP and are now faced with the need for applying XP in a distributed environment. This workshop will be especially valuable to OOPSLA participants who are involved in development activity that is geographically distributed and are interested in applying XP as the methodology in such a setting.

Workshop URL: <http://www.cs.wustl.edu/~mk1/DXP>.

Monday 8:30-17:00

**12 Patterns for Software Architecture**

Jeff Garland, *Crystal Clear Software, Inc.*,  
[jeff@crystalclearsoftware.com](mailto:jeff@crystalclearsoftware.com)  
 Richard Anthony, *Object Computing, Inc.*, [anthony\\_d@ociweb.com](mailto:anthony_d@ociweb.com)  
 Robert Gamoke, *Winona State University*, [rgamoke@winona.edu](mailto:rgamoke@winona.edu)  
 Robert Hanmer, *Lucent Technologies*, [hanmer@lucent.com](mailto:hanmer@lucent.com)

Many authors cite the development of better software architecture as one of the key elements in improving the practice of software development. This is especially true for large-scale systems. Patterns have also emerged as a useful technique to document recurring software designs and the tradeoffs and forces that lead to these designs.

The intent of the workshop is to improve the state of the art and practice for software architecture patterns. Specifically we will discuss how software patterns relate to software architecture, organize the existing architecture patterns in to categories, and identify missing patterns for software architecture. This discussion will be a continuation of an effort begun at ChiliPlop 2002 and available at  
[http://www.crystalclearsoftware.com/cgi-bin/arch\\_patterns/wiki.pl](http://www.crystalclearsoftware.com/cgi-bin/arch_patterns/wiki.pl).

Workshop URL:  
[http://www.crystalclearsoftware.com/cgi-bin/arch\\_patterns/wiki.pl?OOPSLA\\_2002](http://www.crystalclearsoftware.com/cgi-bin/arch_patterns/wiki.pl?OOPSLA_2002).

Monday 8:30-17:00

**14 Agent-oriented methodologies**

John Debenham, *University of Technology, Sydney*,  
[debenham@it.uts.edu.au](mailto:debenham@it.uts.edu.au)  
 Brian Henderson-Sellers, *University of Technology, Sydney*,  
[brian@it.uts.edu.au](mailto:brian@it.uts.edu.au)  
 Nick Jennings, *University of Southampton*, [nrj@ecs.soton.ac.uk](mailto:nrj@ecs.soton.ac.uk)  
 James Odell, *James Odell and Associates*, [mail@jamesodell.com](mailto:mail@jamesodell.com)

Following the success of object technology, the next advance is likely to be the introduction, adoption and widespread use of agent technology for business applications. Agents, building as they do in part on objects, require careful design. Appropriate methodologies for constructing agent-oriented systems may rely to some degree on OO methodologies but their distinct autonomy of an agent means that these agent-oriented processes cannot be as deterministic as they have been in object-oriented developments. Support for emergent processes is required as well as further modifications to existing OO processes. The overall goal of the workshop is to identify a research agenda for the next five years that will enable agent-oriented methodologies to become of "commercial strength" and to be widely adopted by industry.

Workshop URL:  
<http://www.open.org.au/Conferences/oopsla2002/index.htm>.

Monday 8:30-17:00

**15 Tackling the Discovery Costs of Evolving Software Systems**

William Opdyke, *North Central College, Naperville, IL*,  
[opdyke@noctrl.edu](mailto:opdyke@noctrl.edu)  
 Dennis Mancl, *Lucent Technologies*, [mancl@lucent.com](mailto:mancl@lucent.com)  
 Steven Fraser, *Consultant*, [sdfraser@acm.org](mailto:sdfraser@acm.org)

If software is so easy to create, why is it so difficult to change existing software to keep it up to date with changing requirements? Discovery costs - the costs of learning what one needs to know in order to evolve an existing software system - has proven to be the dominant (but often unrecognized) cost in many software systems.

This subject was discussed in a previous OOPSLA panel session: "Do Patterns and Frameworks Reduce Discover Costs?" at OOPSLA '97. This workshop will take a fresh look - five years later - at how organizations are coping with the costs of discovery.

Participants will present and discuss ways of measuring discovery costs, encapsulation and reverse engineering techniques, how best to apply software tools and "agile and extreme" processes to reduce discovery costs, and organizational obstacles.

Workshop URL: <http://csc.noctrl.edu/f/opdyke>.

Monday 8:30-17:00

## 22 J2EE vs .NET

Saeid Tehrani, *EDS Digital Enablement*, saeid.tehrani@eds.com  
 Stephen Jacobs, *EDS Digital Enablement*, steve.jacobs@eds.com  
 Stuart McAlpin, *EDS Digital Enablement*, stuart.mcalpin@eds.com  
 Ali Arsanjani, *IBM National E-business Application Development Center of Competency*, www.arsanjani.org

In the IT industry, innovation happens everyday. This causes challenges for architects, developers, project managers, CIOs and CEOs to constantly learn and evaluate solutions for their projects and their companies.

On one hand, Java has been one of the most successful, innovative and evolving languages/frameworks for the past few years. Many large mainframe applications have been ported to Java. Java started with the slogan of “write once and deploy it many (platforms),” with platforms as small as smartcards, thin and thick clients to large mainframe installations. Nonetheless, Java (J2EE) has become one of the most successful frameworks for the web/application servers.

On the other hand, a new framework called .net has emerged from Microsoft which has many features of J2EE and will be one of the largest competitors of J2EE. Although both frameworks stand on a foundation of programming languages, object models and virtual machines, they are strikingly different when considering the design goals of their runtime environment. The choice of the two dominating platforms will be a hot topic of discussion for the next few years to come.

Workshop URL: <http://www.oopsla2002.freservers.com/>.

Monday 8:30-17:00

## 23 Using Play to Enhance Learning About Objects

William Wake, *Independent Consultant*, William.Wake@acm.org  
 Steven Metsker, *Capital One Financial*, Steve.Metsker@acm.org  
 Pascal Van Cauwenberghe, *Lesire Software Engineering*, pvc@lesire.com

Vera Peeters, *Tryx bvba*, vp@tryx.com

Joshua Kerievsky, *Industrial Logic*, joshua@industriallogic.com

Play is serious! Games, simulations, puzzles, and toys in the classroom can improve student motivation, lead to better long-term memory, and reduce learning anxiety. Used effectively, play can leverage competition and foster collaboration. Above all, play makes learning fun!

We'll explore how play can make it easier to learn about object-oriented technology.

The group will:

- Examine some existing educational games
- Create a framework describing game mechanics
- Explore how different approaches might support learning about OO
- Work in small groups to design new games that teach selected OO topics
- Play-test the new games
- Debrief and critique the games

Who should attend? Anyone who teaches, trains, or wants to learn about instructional play.

To apply, please submit a one- to four-page paper describing your use of games and/or related techniques in teaching object technology.

Workshop URL: <http://www.xp123.com/g4p/workshop>.

Monday 8:30-17:00

## 24 1st International Workshop on Runtime Kernel Support for Dynamic Languages and Component Based Architectures

Yahya Mirza, yahya\_mirza@hotmail.com  
 George Bosworth, *Microsoft Corporation*, georgeb@microsoft.com  
 Jurg Gutknecht, *ETH Zurich*, gutknecht@inf.ethz.ch  
 David Intersimone, *Borland Software Corporation*, david@borland.com  
 Ralph Johnson, *University of Illinois Urbana Champaign*, johnson@cs.uiuc.edu  
 Stan Lippman, *Microsoft Corporation*, slippman@microsoft.com  
 Yukihiro Matsumoto, *ruby-lang.org*, matz@ruby-lang.org

For years, virtual machines have been integral to dynamic languages such as CLOS, Dylan, Smalltalk and more recently Ruby. These systems share facilities including class based objects, garbage collection, dynamic typing, and reflective capabilities. Abstracting these language specific services into a language agnostic runtime will be at the center of the next major platform competitions. To be commercially viable, such a system will need to address a wide range of problems.

The goal of our workshop is to discuss what is required in the kernel of a language agnostic runtime to better support dynamic languages and component based architectures. Our aim is not to focus on the syntactic issues of various programming languages. Instead we emphasize the design and implementation details of an object model generic enough to meet the needs of a wide range of dynamic languages while addressing the problems of component software, mobile computing and the Internet.

Workshop URL:

<http://hosting.msugs.ch/auroraborealis/YMirzaRuntimeKernelOOPSLAWorkshopProposalLong.aspx>.

Monday 8:30-17:00

## 25 “Killer Examples” for Design Patterns and Objects First

Carl Alphonse, *University at Buffalo, State University of New York*, alphonse@cse.buffalo.edu

Philip Ventura, *University at Buffalo, State University of New York*, pventura@cse.buffalo.edu

Dung “Zung” Nguyen, *Rice University*, dxnguyen@cs.rice.edu

Stephen Wong, *Rice University*, swong@cs.rice.edu

The Jargon File defines a killer app as an “application that actually makes a sustaining market for a promising but under-utilized technology.” A killer example provides clear and compelling motivation for some concept. The theme of this workshop is killer examples for design patterns and object-oriented concepts.

Today's computer science students are tomorrow's software developers. They will be expected to design and implement solutions to complex problems. Object orientation is an excellent approach to managing the complexity of large, real-world software systems. Design patterns are an essential part of an object oriented approach to managing complexity. We believe killer examples can motivate students and pique their curiosity about both.

The goal of this workshop is to elicit, share, analyze and critique killer examples from educators and developers. Pre-workshop activities encourage interaction and refinement of examples prior to the workshop.

Workshop URL:

<http://www.cse.buffalo.edu/~alphonse/OOPSLA2002/KillerExamples/>.

## 8:30-12:00 Morning

Monday 8:30-12:00

**5 Extreme Programming Practices in the First CS Courses**

Joe Bergin, *Pace University*, [berginf@pace.edu](mailto:berginf@pace.edu)  
 James Caristi, *Valparaíso University*, [James.Caristi@valpo.edu](mailto:James.Caristi@valpo.edu)  
 Daniel Steinberg, *Dim Sum Thinking*, [dsteinberg@core.com](mailto:dsteinberg@core.com)

Most of the practices of Extreme Programming are beneficial to students in their computer science courses. But in order to teach students properly, pedagogical changes are needed as early as CS1. This workshop seeks participants who have significant ideas for changes that can be made in early computer science courses that involve integrating any of the practices of Extreme Programming or other agile methodologies. Would-be participants should send in a short position paper outlining one or two ideas they have. During the workshop, participants will critically discuss the ideas that have been suggested and explore any new ones that arise. Participants will agree to allow their ideas to be shared via a web page to be posted in various CS educational resources repositories.

Workshop URL: <http://csis.pace.edu/~bergin/XPWorkshop>.

## 13:30-17:00 Afternoon

Monday 13:30-17:00

**3/ Expanding the Boundaries of Unit Testing**

Joseph Pelrine, *MetaProg*, [jpelrine@metaprogram.com](mailto:jpelrine@metaprogram.com)  
 Steve Freeman, *Thompson Financials*, [steve@m3p.co.uk](mailto:steve@m3p.co.uk)  
 Tim MacKinnon, *Connextra*, [tim@connextra.com](mailto:tim@connextra.com)

One of the major benefits of the Agile Development movement has been to move unit testing to its rightful place as the software development technique. It has also supplied us with a family of unit testing frameworks to assist with this task. Unit testing, however, takes effort and skill to do well, especially for applications such as distributed systems.

This workshop is concerned with bringing together and sharing experience between people who have experience of unit testing of software, particularly for Test-Driven Development.

Participants will be asked to submit a position paper detailing their experience with unit testing, and illustrating what they've done to expand the boundaries of unit testing. Typical examples would include extensions to the xUnit testing framework, automating tests etc. The organizers will accept position statements based on originality, relevance and suitability for discussion.

Please mail your submissions (preferably in PDF) to Joseph Pelrine ([jpelrine@metaprogram.com](mailto:jpelrine@metaprogram.com)) by September 15.

Workshop URL: <http://www.metaprogram.com/OOPSLAWorkshop>.

## Tuesday, 5 November

8:30-17:00 Full day

Tuesday

8:30-17:00

### 4 Generative Techniques in the context of Model-Driven Architecture

Jorn Bettin, *SoftMetaWare, New Zealand*, [joern.bettin@acm.org](mailto:joern.bettin@acm.org)  
 Ghica van Emde Boas, *Bronstee.com Software & Services, The Netherlands*, [emdeboas@bronstee.com](mailto:emdeboas@bronstee.com)

Craig Cleaveland, *Independent software consultant, USA*,  
[craig@craigc.com](mailto:craig@craigc.com)

Krzysztof Czarnecki, *DaimlerChrysler Research and Technology, Germany*, [krzysztof.czarnecki@web.de](mailto:krzysztof.czarnecki@web.de)

Model-driven Architecture (MDA) is an initiative by the OMG to leverage UML-based modeling techniques to insulate abstract software system specifications from implementation dependencies. The workshop focuses on generative techniques that can be used to realize MDA. The aim is to bring together practitioners, researchers, academics, and students to discuss their experience with generative techniques and MDA.

Topics of interest include:

- synergy between MDA, components and generative techniques;
- designing domain specific languages on the basis of UML;
- modeling variability within product lines;
- model-to-model transformations;
- styles of model-driven generators;
- model-driven template languages (design, execution environment, debugging, editors, management of template code);
- specification of heuristics and manual design decisions
- use of XSLT for MDA;
- generation of code & non-code artifacts;
- influence of MDA on software architecture;
- MDA and agile development;
- industrial applications of MDA.

The goal is to share experience, consolidate successful techniques, and identify the most promising application areas and open issues for future work. Tracks: Emerging Technologies; Model Driven Architecture, Languages

Workshop URL:

<http://www.softmetaware.com/oopsla2002/mda-workshop.html>.

Tuesday

8:30-17:00

### 6 Architecture is dead - Long live the Architect!

Klaus Marquardt, *Dräger Medical AG*, [marquardt@acm.org](mailto:marquardt@acm.org)

Alan O'Callaghan, *De Montfort University*, [aoc@dmu.ac.uk](mailto:aoc@dmu.ac.uk)

Jens Coldewey, *Coldewey Consulting*, [jens\\_coldewey@acm.org](mailto:jens_coldewey@acm.org)

Most approaches to software architecture focus on the initial up-front activities needed in large projects. In contrast, the agile manifesto states that "the best architectures [...] emerge from self-organizing teams." This suggests that most of the architecture is defined while the system is already under development. The roles and tasks related to software architecture need to be rethought: What does the architect actually do while the project is on its way?

This workshop takes a close look at the role of the software architect, and the related responsibilities and tasks. The participants will explore how changes to the process and project culture influence this role, and what aspects of the role remain stable throughout different processes. The objective is to collect practices and working attitudes that help the agile project team to deal with software architecture, and the practicing architect to deal with agile projects.

Workshop URL:

<http://www.kmarquardt.de/workshops/ArchitectureInAgileProjects.htm>.

Tuesday

8:30-17:00

### // Patterns in Distributed Real-Time and Embedded Systems

Michael Kircher, *Siemens AG, Corporate Technology, Munich, Germany*, [Michael.Kircher@mchp.siemens.de](mailto:Michael.Kircher@mchp.siemens.de)

Prashant Jain, *Manystreams, Inc., India*,

[prashant@noida.manystreams.com](mailto:prashant@noida.manystreams.com)

Douglas C. Schmidt, *University of California, Irvine, USA*,  
[schmidt@uci.edu](mailto:schmidt@uci.edu)

Aniruddha Gokhale, *Vanderbilt University, USA*,  
[gokhale@isis-server.vuse.vanderbilt.edu](mailto:gokhale@isis-server.vuse.vanderbilt.edu)

Distributed real-time and embedded (DRE) systems are becoming ubiquitous, not just in the aerospace domain, but also in telecommunications, automotive, and process control. As software replaces functionality traditionally provided by hardware, better software development technologies are needed to ensure end-to-end system quality of service properties and to control lifecycle costs.

Increasingly, object-oriented (OO) technologies are being considered as the underpinning for mission-critical DRE systems. Because OO technologies were not initially intended for the time and space constraints of DRE systems, researchers and practitioners often struggle to apply OO techniques effectively so their tasks become easier, not harder.

What are common patterns and practices that can be applied to overcome many recurring challenges and constraints related to developing DRE systems? The goal of this workshop is to bring together researchers and practitioners from the field of DRE systems to begin documenting a common "pattern language" for DRE systems.

Workshop URL: <http://www.cs.wustl.edu/~mk1/RealTimePatterns>.

Tuesday 8:30-17:00

**13 Patterns for Customer Interaction and Expectation Management**

Linda Rising, *Independent*, [risingl@acm.org](mailto:risingl@acm.org)  
 Bill Opdyke, *North Central College, Naperville, IL*, [opdyke@noctrl.edu](mailto:opdyke@noctrl.edu)  
 Terry Fujino, *InArcadia*, [pv@inarcadia.co.jp](mailto:pv@inarcadia.co.jp)  
 Tina Nakatani, *Wakayama University, Japan*, [tina@s-lagoon.co.jp](mailto:tina@s-lagoon.co.jp)  
 Masao Tomono, *Exa Corp.*, [mtomono@acm.org](mailto:mtomono@acm.org)  
 David DeLano, *OTI*, [delanod@acm.org](mailto:delanod@acm.org)

For companies to survive, they must satisfy the needs of their customers. Insights into understanding and managing customer expectations come through experience and patterns are a great means for capturing and leveraging these insights.

This goal of this workshop is to expand on previously published patterns for customer interaction by Linda Rising and continued at ChiliPLoP 2002. A pattern language for Customer Interaction and Expectation Management will be the end product. This pattern language should be useful for both object-oriented developers and management. eXtreme Programming and other agile approaches stress the importance of a close working relationship with the customer. Preferably, the customer should be on-site and receive releases often to build a trusting relationship between the development team and the customer representative. Like many good guidelines, this one from the agile community needs fleshing out, especially for some developers who have no history of working closely with customers.

Workshop URL: <http://www.lindarising.org>.

Tuesday 8:30-17:00

**16 1st Workshop on Software Development Process Patterns**

Michael Gnatz, *Technische Universität München, Germany*,  
[gnatzm@in.tum.de](mailto:gnatzm@in.tum.de)  
 Frank Marschall, *Technische Universität München, Germany*,  
[marschal@in.tum.de](mailto:marschal@in.tum.de)  
 Gerhard Popp, *Technische Universität München, Germany*,  
[popp@in.tum.de](mailto:popp@in.tum.de)  
 Andreas Rausch, *4Soft GmbH, Germany*, [rausch@4soft.de](mailto:rausch@4soft.de)  
 Wolfgang Schwerin, *Technische Universität München, Germany*,  
[schwerin@in.tum.de](mailto:schwerin@in.tum.de)

In industrial practice a flexible and modular process model is needed that integrates the benefits of the various existing process models and can be customized to the specific needs of development projects. Therefore a comprehensive metamodel for software processes based on a clear notion of process artefacts, like for instance the concept of Process Patterns, is needed. With this workshop we aim to foster a common understanding of a software development process metamodel and on description techniques for concrete development processes and their artefacts based on this metamodel. Such a common Process Pattern language might build the general basis for the integration and evolution of process knowledge from different software engineering communities.

Workshop URL: <http://www.forsoft.de/zen>.

Tuesday 8:30-17:00

**17 Built for Life: Constructing Software to Outlive Its Creators**

Andrew Hunt, *The Pragmatic Programmers*,  
[andy@pragmaticprogrammer.com](mailto:andy@pragmaticprogrammer.com)  
 Brian Marick, *independent consultant*, [marick@visibleworkings.com](mailto:marick@visibleworkings.com)  
 Andy Schneider, *BJSS*, [andy.schneider@bjss.co.uk](mailto:andy.schneider@bjss.co.uk)  
 Dave Thomas, *The Pragmatic Programmers*,  
[dave@pragmaticprogrammer.com](mailto:dave@pragmaticprogrammer.com)

At some point, successful software leaves the care of its creators and ventures into the world on its own. There, it will have to be changed and understood by others. In this workshop, we will explore techniques for making software more easily understood by programmers, testers, and even end users inclined to tinker. What can we do \*now\* that will help them \*then\*?

For better or worse, those who inherit software often do not inherit design or architecture documents. Or what they get is out of date or inaccurate. Since code is less likely to get lost, we focus on code and, secondarily, on tests.

Possible topics include:

- tracing and logging
- tests as documentation
- breaking encapsulation to aid understanding
- representation of design decisions and alternatives in code
- intention-revealing error handling
- self-diagnostic code
- how to avoid leaving misleading cues in the code

Workshop URL: <http://visibleworkings.com/built-for-life/>.



Tuesday 8:30-17:00

**18 Semantics of Enterprise Integration II**Mark Lycett, *Brunel University*, [mark.lycett@brunel.ac.uk](mailto:mark.lycett@brunel.ac.uk)Paul Allen, *Computer Associates*, [paul.allen@ca.com](mailto:paul.allen@ca.com)Sergio de Cesare, *Brunel University*, [sergio.decesare@brunel.ac.uk](mailto:sergio.decesare@brunel.ac.uk)Grant Holland, *Sun Microsystems*, [Grant.Holland@sun.com](mailto:Grant.Holland@sun.com)Chris Simons, *Aonix Europe Ltd.*, [c.simons@aonix.co.uk](mailto:c.simons@aonix.co.uk)

Business systems integration has become increasingly significant, requiring growing levels of investment and resources. The integration of such systems, however, is not just technical in nature but spans the interrelated dimensions of people, organization and technology. The primary challenge concerns effective communication and coordination within and across these dimensions and the definition of common business semantics provides a key issue.

Current approaches to semantic interoperability include ontologies, services and agency. Ontology-based approaches seek to define common domain terminologies. Service-based approaches concentrate on the specification of business capability and behavior. Agency-based approaches provide mechanisms for negotiation and resolution of conflicts. Such approaches are coupled with emerging and consolidated technologies (e.g. XML) that have the potential of providing effective solutions to the problem of enterprise integration.

This workshop is aimed at facilitating and spawning discussion on the issues surrounding enterprise integration. More specifically, participants will be invited to:

- Identify key obstacles in relation to semantic interoperability of systems;
- Improve the State-of-the-Art in the definition and representation of the semantics in enterprise and systems integration;
- Instigate collaborative research efforts among the participants.

This workshop is addressed to all academics, researchers and practitioners interested in approaches aimed at achieving the integration of enterprise systems at all levels.

Workshop URL:

[http://www.fluidbusiness.org/events/sei2002\\_workshop/](http://www.fluidbusiness.org/events/sei2002_workshop/).

Tuesday 8:30-17:00

**19 Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE)**Vinny Cahill, *Trinity College, Dublin*, [vinny.cahill@cs.tcd.ie](mailto:vinny.cahill@cs.tcd.ie)Siobhan Clarke, *Trinity College, Dublin*, [siobhan.clarke@cs.tcd.ie](mailto:siobhan.clarke@cs.tcd.ie)Jim Dowling, *Trinity College, Dublin*, [jim.dowling@cs.tcd.ie](mailto:jim.dowling@cs.tcd.ie)Robert Filman, *NASA Ames Research Center*, [rfilman@arc.nasa.gov](mailto:rfilman@arc.nasa.gov)

Object technology is becoming the technology of choice for programming ubiquitous computing devices and their environment. Different flavours of Java and middleware are appearing to allow the interconnection and interaction between pervasive devices operating at different scales. Within this environment, context is an important and powerful concept, referring to the physical and social situation in which objects are embedded.

If context-aware applications and environments are to be built, what new set of issues must be confronted? This workshop is intended to bring together researchers and practitioners to discuss the issues inherent in engineering context-aware applications and environments. The goal of this workshop is to determine those issues in representing, interpreting and delivering context information that can help lead to new approaches and techniques for building a generic supporting infrastructure for object-oriented context-aware applications.

Workshop URL: <http://www.dsg.cs.tcd.ie/ecoose/oopsla2002/>.

Tuesday 8:30-17:00

## 20 Object-Oriented Web Services

Peri Tarr, *IBM Thomas J. Watson Research Center*,  
tarr@watson.ibm.com

Anthony Finkelstein, *University College London*,  
a.finkelstein@cs.ucl.ac.uk

Brent Hailpern, *IBM Thomas J. Watson Research Center*,  
bth@watson.ibm.com

Giacomo Piccinelli, *Hewlett-Packard Labs, Bristol*,  
giapicc@hplb.hpl.hp.com

Judith Stafford, *Software Engineering Institute, Carnegie Mellon University*, jas@sei.cmu.edu

Web services are software that provide access to their capabilities across the World Wide Web via XML interfaces. The use of XML enables systems running in different environments and at different locations to exchange information, interoperate, and be combined more readily than ever before. The first generation of web-service infrastructure and applications are already in development, allowing web services to issue requests to others and to register/describe/find a service to use (e.g., SOAP, WSDL, UDDI).

Eventually, web services will become electronic utilities that are delivered to end users over the Internet-representing a critical new application domain in electronic commerce. Like traditional utilities, such as telephone and electricity, web services will be metered and customers will pay for their use. The terms of use (called service level agreements or SLAs) of web services will include functionality, performance, and reliability. The need to dynamically monitor and control SLAs and to respond to changing needs and resources imposes challenging requirements on the design, development, deployment, and evolution of web services.

This workshop will explore issues in, and new technologies and methodologies to support the engineering, deployment, integration, and evolution of web services in such a way as to get the best software engineering properties of objects into XML-based web services.

Workshop URL:

<http://www.research.ibm.com/people/b/bth/OOWS2002.html>.

Tuesday 8:30-17:00

## 21 Software Apprenticeship: the Journey to Mastery

Ken Auer, *RoleModel Software, Inc.*, kauer@rolemodelsoft.com

Nathaniel Talbott, *RoleModel Software, Inc.*,  
ntalbott@rolemodelsoft.com

Pete McBreen, *McBreen Consulting*, petemcbreen@acm.org

Robert Martin, *Object Mentor, Inc.*, rmartin@objectmentor.com

Apprenticeship is an age-old concept, while software as we know it is very young. This workshop will explore the application of the apprenticeship model to the training of software developers. It will not attempt to introduce or defend apprenticeship, as it is expected that participants will already have a working knowledge of the concept and believe that at least parts of it can be practically applied. We will work together to refine the definition of what software apprenticeship is, how it can be applied today, and how it ought to grow and mature.

Workshop URL:

<http://www.rolemodelsoft.com/OOPSLA2002/apprenticeship>.

Tuesday 8:30-17:00

## 26 Using Domain Specific Languages to Drive Business Applications

Ali Arsanjani, *IBM, E-business Application Development, Center of Competency*, arsanjan@us.ibm.com

Reza Razavi, *University of Paris, LIP6*, razavi@acm.org

The need for rapid adaptation of domain applications to ever changing business models has brought industrials to build runtime adaptable, "expert-programmable" software using home-made techniques and tools. Ongoing research on Adaptive Object-Models and Best-practices in Business Rule Design and Implementation shows that in recurrent cases such applications are built themselves as domain-specific languages, ensuring in this way runtime programmability (and therefore adaptability) of the software.

The goal of this workshop is to provide a platform for researchers and practitioners interested in dynamically adaptable business software implemented as a domain-specific language, and related approaches like Grammar-oriented Object Design, to exchange ideas and experience on how to systematize building these applications. We will try to establish a catalogue of different techniques used and the conditions under which they work better. We would also like to compare design techniques that are used for creating these applications, that allow often runtime class specialization by non-programmers, with that of object-oriented languages, themselves.

Workshop URL: <http://www.arsanjani.org/oopsla2002>.

Tuesday 8:30-17:00

## 27 Extravagaria - Art Assisting Science

Richard Gabriel, *Sun Microsystems, Inc.*, rpg@dreamsongs.com

Joseph Bergin, *Pace University*, berginf@pace.edu

Artists and scientists share the most important goal: the search for truth. Their methods and approaches differ, and their results are in different languages, but there is no a priori reason to grant primacy to either. In educational research, and possibly in other domains, there is thriving work on using the techniques of art - especially creative writing - to explore research questions and explain results. In software, we not only seek truth and reality, but we create it. Certainly there is a role for art here. This workshop explores that question.

Workshop URL:

<http://www.dreamsongs.com/Feyerabend/Extravagaria.html>.

**Tuesday 8:30-17:00**

## 28 Agile Processes Workshop II - Managing Multiple Agile Projects

Mike Beedle, *Hipaa Accelerator Inc.*, [beedlem@e-architects.com](mailto:beedlem@e-architects.com)  
 Ken Schwaber, *Advanced Development Methods*, [virman@aol.com](mailto:virman@aol.com)  
 Jeff Sutherland, *PatientKeeper*, [jeff.sutherland@computer.org](mailto:jeff.sutherland@computer.org)  
 John Cerwin, *Hipaa Accelerator Inc.*, [cerwinj@hipaaccelerator.com](mailto:cerwinj@hipaaccelerator.com)

The goal of this workshop is to understand and contrast the new enterprise-level agile processes in existence today such as: Scrum, XBreed, XP@Scrum and Grizzly. A detailed examination of each enterprise-level agile process will be conducted to understand its practices. Case studies for each process will be provided as empirical evidence. Similarities, differences, synergies and misfits among different enterprise-level agile processes and practices will be examined both from the practice and theory perspectives.

In addition, discussions of how to introduce enterprise-level agile methods, justify them in terms of ROI, ensure the completion of successful projects, conduct training and mentoring, and transition from non-agile to enterprise-level agile methods will be conducted.

Workshop URL: <http://www.hipaaccelerator.com/oopsla2002>.

**Tuesday 8:30-17:00**

## 30 Product Line Engineering - The early steps: Planning, Managing, and Modeling (PLEES'02)

Klaus Schmid, *Fraunhofer IESE*, [schmid@iese.fhg.de](mailto:schmid@iese.fhg.de)  
 Birgit Geppert, *Avaya Research Labs*,  
[bgeppert@research.avayalabs.com](mailto:bgeppert@research.avayalabs.com)

Product line engineering is a new and ambitious approach, aiming at developing a whole set of products in an integrated manner. Product line development approaches usually use architectural approaches, component technologies and OO-frameworks as a basis for the high level of reuse among familiar members. While these technologies have been studied quite extensively, technology by itself is insufficient for successfully changing an organization into a product line development organization. In this workshop we will focus on the early stages of product line engineering like economic planning, management, and modeling of product line development and the organizational embedding of the shift.

It is our aim to establish a forum for the successful exchange among product line practitioners and researchers. A key goal of the workshop will be to establish a preliminary research agenda and compile lessons learned from the field.

Workshop URL: <http://www.plees.info>.

**8:30-12:00 Morning**

**Tuesday 8:30-12:00**

## 29 Pair Programming Explored

Steven Fraser, *Consultant*, [sdfraser@acm.org](mailto:sdfraser@acm.org)  
 Laurie Williams, *CS - NC State*, [williams@csc.ncsu.edu](mailto:williams@csc.ncsu.edu)

Pair Programming is one of the XP software best-practices. It is also a stand-alone practice. This workshop will bring together both Pair Programmers and researchers studying pair programming to evaluate the effectiveness of the practice and avenues for improvement.

Workshop URL: <http://collaboration.csc.ncsu.edu/pairprogramming>.

# DESIGNFEST

**Chair: Gail E. Harris, Instantiated Software Inc.,  
Gail.Harris@Instantiated.CA**



The most talked about events at past OOPSLA's have typically been the ones with audience involvement. Well, at DesignFest, this is taken to the extreme. DesignFest is not about passively sitting and listening to experts talk about design. DesignFest is about sharpening your design skills by rolling up your sleeves and working on a real problem with others in the field. You can expect to interact with everyone from total beginners to textbook authors!

The DesignFest is a free event (for conference registrants) that was created to give OOPSLA attendees the opportunity to learn more about design by doing it. DesignFest is an increasingly popular event at OOPSLA and many participants choose to return year after year. If you don't have much knowledge about design, this is a great way to get some first-time experience. If you are experienced, DesignFest is the perfect place to try out new ideas, and see how your peers react.

In the DesignFest, you will work in small groups to solve a particular design problem, bringing to bear your experience and skills in object-oriented design and/or experience working on similar problems. The goal is to learn new techniques from each other and to uncover and articulate the analysis and design patterns that we already use subconsciously.

**NEW THIS YEAR** will be Extreme Programming teams during DesignFest. So bring your laptop! A small number of teams will be able to work on a problem using Extreme Programming practices.

In recent years, the CodeFest teams have become a regular part of DesignFest. These teams of student programmers will implement some of the designs created during the DesignFest. In this way the DesignFest participants will be able to see how good their designs really are. OOPSLA will sponsor a small number of student teams of three persons each to participate in CodeFest. If you are a student and have a team proposal from your school, please contact the DesignFest Chair at [designfest@oopsla.acm.org](mailto:designfest@oopsla.acm.org).

Starting in 1999, AnalysisFest was added to DesignFest. In these sessions, open to DesignFest participants, professionals form groups to carefully study statements of need and requirements documents. Some of these are the same documents used in DesignFest, but others are not. Each group must come up with precise specifications of the domain of the problem they are given, without any regard to solution (design). AnalysisFest is not a tutorial on object-oriented analysis; it is an opportunity for analysts to sharpen and measure their skills by interacting with their peers.

The problem descriptions will be distributed at the start of each DesignFest session. DesignFest teams will also be provided with a timetable, process hints, and a list of deliverables. Each team will be asked to produce a poster or two that summarizes their work. These posters will be presented in the exhibit area for all conference attendees to view.

Each team will have a moderator and a recorder. The moderator is responsible for making sure that the team makes progress, without leading them in any particular direction. The recorder is responsible for recording the design the group produces and what they have learned.

The following problems have already been selected for this year's DesignFest:

- **My Family Reunion** - a web site for interacting with your family.
- **eCommerce** - a web shopping site including catalogue management
- **Enterprise Storage** - a reliable, robust data storage and backup utility
- **Bidding** - a web service for customers to submit requirements and contractors to submit bids
- **Supply Chain** - an enterprise system to manage the supply chain of a large retailer with geographically distributed, franchised outlets
- **Robotic Motion Control in a Restricted Domain** - a motion control system for a robotic arm that allows it to safely and efficiently move to any point in a restricted domain

To register, please fill out the DesignFest section of the OOPSLA registration form. Registration for DesignFest is free if you have registered for the conference. Be sure to include your e-mail address. You will be contacted by e-mail at a later point in time with regard to selecting a design problem, preference to work on an XP team, your preference to do analysis or design, and possibly a team meeting time.

You are invited to take a tour of the previous DesignFests and CodeFests at <http://designfest.acm.org>.

The DesignFest Committee appears in the Conference Committees section of the Program.

Note that all DesignFest and AnalysisFest sessions are separate sessions, lasting half a day or a full day, while the CodeFest is a continuous event.

## Monday, 4 November

**8:30-17:00**      **Full Day DesignFest**  
**12:00-17:00**      **CodeFest**

## Tuesday, 5 November

**8:30-17:00**      **CodeFest**

## Wednesday, 6 November

**10:00-17:00**      **CodeFest**  
**10:30-17:00**      **Full Day DesignFest**  
**13:30-17:00**      **Half Day DesignFest**

## Thursday, 7 November

**10:00-17:00**      **CodeFest**  
**13:30-17:00**      **Half Day DesignFest**

## Friday, 8 November

**10:00-13:00**      **CodeFest**  
**17:00-18:00**      **DesignFest/CodeFest Wrap-Up**

# EDUCATORS' SYMPOSIUM

**Chair: Helen Sharp, The Open University, UK,  
h.c.sharp@open.ac.uk**



The Educators' Symposium provides a forum for object-oriented educators, both academic professors and industrial trainers, to meet and exchange ideas, experiences and resources. We have a programme of submitted posters and papers, which is set out below, but all attendees are encouraged to bring along their ideas and challenges, to pin them up alongside the posters, and to share their insights, frustrations and solutions with other attendees. You'll see that

there is an hour put aside for viewing and discussing ideas displayed as posters, so there's plenty of time to obtain feedback. We will also hear reports from other conferences, workshops and projects related to object-oriented education that have happened throughout the year. But don't think you'll just be sitting back and listening. There are activities for you to join in with: a design review, and an invited session exploring activities to help you teach design patterns! We'll also be expanding your horizons by re-examining the way use cases are used and taught, and whether OO education has failed, and if so why? Please come and join us for a stimulating and thought-provoking day, in the best tradition of OOPSLA.

Participation in the Educators' Symposium requires separate registration. Please refer to the registration form on *page 105*.

**Tuesday 8:30-10:00**

## Welcome and Opening Talk

### Welcoming remarks

Helen Sharp, *The Open University, UK, h.c.sharp@open.ac.uk*

### Nearly the end of the line for use cases?

Richard Mitchell, *InferData Corporation, richard@inferdata.com*

Many analysts find use cases a comfortable way to capture requirements, and many teachers give them a central role in courses based on UML. However, many developers report that use cases don't tell them what they need to know to develop systems. Perhaps use cases are too comfortable? Of course, use cases are different things to different people. Commonly, analysts use them as their originator intended, to express sequences of interactions between users and systems. Used this way, use cases are partly about the functionality of a system, and partly about delivering this functionality to users. Separating these two issues can yield models that are more informative. This talk will show you one way of making the separation, in which use cases are about delivering functionality.

**Tuesday 10:30-11:30**

## Paper session: Teaching XP

### *Pair Programming in an Introductory Computer Science Course: Initial Results and Recommendations*

Laurie Williams, *North Carolina State University, williams@csc.ncsu.edu*

Kai Yang, *North Carolina State University, kyang@unity.ncsu.edu*

Eric Wiebe, *North Carolina State University, wiebe@unity.ncsu.edu*

Miriam Ferzli, *North Carolina State University, ferzli@unity.ncsu.edu*

Carol Miller, *North Carolina State University, miller@unity.ncsu.edu*

### *Exploring Pair Programming in Distributed Object-Oriented Team Projects*

Prashant Baheti, *North Carolina State University, ppbaheti@unity.ncsu.edu*

Laurie Williams, *North Carolina State University, williams@csc.ncsu.edu*

Edward Gehringer, *North Carolina State University, efg@ncsu.edu*

David Stotts, *University of North Carolina at Chapel Hill, stotts@cs.unc.edu*

### *A Course in Software Engineering Built Around an eXtreme Programming Project*

Daniel H. Steinberg, *Dim Sum Thinking, Inc., DSteinberg@core.com*

Daniel W. Palmer, *John Carroll University*

### *What I Learned Teaching XP*

Ivan Tomek, *Acadia University, ivan.tomek@acadiau.ca*

**Tuesday 11:30-12:00**

## Conference Reports

In this session, we will hear reports from OO education-related conferences that have happened throughout the year

**Tuesday 13:00-13:45**

## Paper Session: Teaching Design

### *Learning Appreciation for Design Patterns by Doing it the Hard Way First*

Dale Skrien, *Colby College, Waterville, djskrien@colby.edu*

### *Role Playing: Easing the Paradigm Shift*

Steven K. Andrianoff, *St. Bonaventure University, ska@cs.sbu.edu*

David B. Levine, *St. Bonaventure University, dlevine@cs.sbu.edu*

Joseph Bergin, *Pace University, berginf@pace.edu*

### *Experiences Using Design-n-Code Fests as Capstone Projects for an Object-Oriented Software Development Course*

Stephen W. Clyde, *Utah State University, swc@cs.usu.edu*

Angelique E. Crane, *Utah State University, angel@mdsc.com*



**Tuesday 13:45-14:30**

### DesignFest Review

This session will involve a review of a design produced during a DesignFest activity earlier in the conference week. Attendees will be introduced to the problem, and the designed solution, and they will be able to question members of the design team who produced it.

**Tuesday 14:30-15:30**

### Break and Poster Time

In this session, attendees will be encouraged to wander around and view the posters and other teaching materials people have brought to share. Some posters have been pre-selected (listed below) while others may be brought along on the day by other attendees

#### *Tracking the Design of Objects*

Adair Dingle, *Seattle University*, [dingle@seattleu.edu](mailto:dingle@seattleu.edu)

#### *Jed Development Notes*

J.R.Fisher, *California State Polytechnic University*,  
[jrfisher@csupomona.edu](mailto:jrfisher@csupomona.edu)

#### *Teaching Object Concepts for XML-based Representations*

Robert L. Kelsey, *Los Alamos National Laboratory*, [rob@lanl.gov](mailto:rob@lanl.gov)

#### *Design Patterns for Self-Balancing Trees*

Dung Nguyen, *Rice University, Houston*, [dxnguyen@rice.edu](mailto:dxnguyen@rice.edu)  
Stephen B. Wong, *Rice University, Houston*, [swong@rice.edu](mailto:swong@rice.edu)

#### *Design Patterns in the Standard Template Library*

Arturo Sanchez, *University of North Florida*, [arturo@acm.org](mailto:arturo@acm.org)

#### *Teaching Reflective Architectures*

Sara Stoecklin, *Florida State University*, [stoeckli@cs.fsu.edu](mailto:stoeckli@cs.fsu.edu)  
Judy Mullins, *University of Missouri-Kansas City*, [mullinsj@umkc.edu](mailto:mullinsj@umkc.edu)

#### *Teaching Data Structures Using Object-Oriented Toolkits*

Josh Tenenberg, *University of Washington, Tacoma*,  
[jtenenbg@u.washington.edu](mailto:jtenenbg@u.washington.edu)

**Tuesday 15:30-16:00**

### Project and Workshop Reports

In this session, we will hear reports from OO education-related projects and workshops that have happened throughout the year

**Tuesday 16:00-17:00**

### Activity

Jane Chandler, *Portsmouth University*, [jane.chandler@port.ac.uk](mailto:jane.chandler@port.ac.uk)

This session will introduce a number of games designed for teaching the underpinnings and basics of Patterns. Games provide a practical, activity-based way of introducing challenging or difficult topics in a non-threatening way while enabling students/trainees to link new material into a familiar framework and learn from their experiences. In addition games can be structured to enable students who have some familiarity with the concept of Patterns to use their knowledge in a practical way that increases their depth of understanding. Delegates will have the opportunity to try one or more games and then consider ways of adapting these and other games to use in their own teaching and training.

**Tuesday 17:00-17:30**

### Closing Plenary

#### *The Failure of Object Education: introduction to the debate*

David West, *New Mexico Highlands University*

#### *Open Mike and Closing Remarks*

Helen Sharp, *The Open University, UK*, [h.c.sharp@open.ac.uk](mailto:h.c.sharp@open.ac.uk)

## DOCTORAL SYMPOSIUM

**Chair: James Noble, Victoria University of Wellington, New Zealand, [kjx@mcs.vuw.ac.nz](mailto:kjx@mcs.vuw.ac.nz)**



The 2002 OOPSLA Doctoral Symposium is preparing the leaders of the future of object-oriented technology. In the symposium, a selected group of doctoral students present their work and obtain guidance from university and industry mentors and other symposium participants. The goal of the symposium is to expose students to helpful criticism before their thesis defense and to provide advice on future research and employment.

We encourage doctoral students who have established a thesis topic to apply to present their work at the symposium (see the Call for Participation on the OOPSLA web page). As in the past, this year's selected participants will be chosen on the basis of the mentors' evaluation of the students' descriptions of their research, and whether the work was sufficiently advanced to have some preliminary results but with sufficient time remaining to benefit from the symposium experience.

Mentors have traditionally been selected from respected individuals in the OOPSLA community, such as established researchers, members of the program committee, workshop leaders, etc. Their mission is not only to be constructively critical about the current status of the work, but also to give advice to where the work should focus and stress the future direction for successful completion of the doctoral work. This year's mentors are Prof. Alan Borning (The University of Washington), Dr. Dirk Riehle, Dr. Mira Mezini (University of Siegen and Northeastern University), Dr. Mario Wolczko (Sun Microsystems Laboratories Inc.) and Dr. James Noble (Victoria University of Wellington).

Due to the mentoring nature of the event, the symposium itself will only be open to those selected for participation. So OOPSLA attendees can meet the students and discuss their research, all doctoral symposium presenters will have a poster on display during the conference and a two-page short paper published in the Conference Companion proceedings. These provide opportunities for additional feedback and suggestions on the dissertation work, contacts for further interaction, and experience in communicating with other professionals. Each student also receives a certificate of participation and support to attend the conference.

## DEMONSTRATIONS

**Chair: Ken Bauer, ITESM, Campus Guadalajara, [kenbauer@acm.org](mailto:kenbauer@acm.org)**



OOPSLA demonstrations provide an opportunity for companies and universities to show their latest work to an experienced audience. This can be work in progress, commercial applications, proof of concepts, results of academic research, tools, systems or any topic that has interesting object-oriented aspects. Demonstrations are not focused on selling a product but to highlight, explain and present the technical aspects of it. Demonstrators may actively solicit feedback from

the usually very technically savvy audience. In the past, this has made for some very interesting demonstration sessions.

The demonstrations will be held Wednesday, 6 November 2002 and Thursday, 7 November 2002. Each session lasts up to 45 minutes and several sessions run in parallel. This year your demonstrations will be held in rooms constructed inside the exhibit hall to attract many people.

Persons or groups that work with an interesting topic, suitable for demonstrating to a group of 30 to 100 people, should use this opportunity to show their work to a larger audience. Take this opportunity to demonstrate your work to an audience that is for once not just interested **what** it does, but really wants to know **HOW** it works!

Accepted demonstrations may also (and are encouraged to) provide a two-page summary description to be published in the OOPSLA 2002 Conference Companion.

Proposals for demonstrations will continue to be accepted on quality, suitability and space available basis. However, in order to be included in the final program they must be submitted before 19 July 2002. Submission is via the online submission system only. Visit <http://oopsla.acm.org/demonstrations/submit/> for instructions on how to submit a demonstration. The demonstrations chair can be contacted for further questions at [demos@oopsla.acm.org](mailto:demos@oopsla.acm.org).

## POSTERS

**Chair: Laurie Williams, North Carolina State University, [williams@csc.ncsu.edu](mailto:williams@csc.ncsu.edu)**



OOPSLA 2002 Posters cover the same interest areas as the Technical Papers, Practitioner Reports and the Onward! program. The poster session is an informal and highly interactive environment that gives OOPSLA attendees the opportunity to engage with one another in discussions about relevant, ongoing work and critical issues in key areas. The session also give conference attendees the chance to learn about work in areas with which they want to become familiar and about

preliminary research results. Researchers and practitioners can describe their work-in-progress and elaborate on work presented in other conference forums. All OOPSLA participants can obtain rapid, low-cost introductions to interesting work and technologies in object-oriented software engineering. They can also provide input and feedback directly to the authors.

The Posters program begins with a special session at the Welcome Reception on Tuesday evening. All posters will be on display and the authors will be present to meet with attendees and discuss their work. Poster also enable interactions to occur throughout the conference. After the Welcome Reception, the posters will be displayed in the Exhibit Hall, so that OOPSLA participants can view them at their convenience. Poster authors will be available as time permits. The goal is to encourage small groups of individuals interested in a technical area to gather and interact.

Posters submission deadline is **19 July 2002**. Visit the Poster Sessions Call for Participation page ([http://oopsla.acm.org/2f\\_posters.html](http://oopsla.acm.org/2f_posters.html)).

## THE JOINT ACM SIGPLAN STUDENT RESEARCH COMPETITION AND OOPSLA POSTER SESSION

**Chair: Laurie Williams, North Carolina State University, [williams@csc.ncsu.edu](mailto:williams@csc.ncsu.edu)**

### An exciting, new opportunity

The OOPSLA Poster Session has been extended to include the first ACM SIGPLAN Student Research Competition. The Poster Session is an excellent forum for presenting research in progress in order to obtain feedback from conference participants. The ACM SIGPLAN Student Research Competition shares the common goal of having students meet and interact with researchers so that both sides have the opportunity to learn of ongoing, current research. The Student Research Competition has the additional goal of the student gaining experience *with both formal presentations and evaluations*.

Students wishing to participate in the Student Research Competition must submit a two-page abstract and a poster outline, following the regular Poster Session guidelines. These requirements can be found on the Poster Session Call For Participation page ([http://oopsla.acm.org/2f\\_posters.html](http://oopsla.acm.org/2f_posters.html)). However, when submitting their poster materials, interested students must indicate their desire to participate in the competition. The same applies to the Doctoral Symposium applicants if they wish to be considered for this competition. Submissions will be reviewed by a committee, and ten students will be chosen to participate in the competition. These students will discuss their work with competition judges (and conference attendees) during the regular Poster Session. The Poster session is used to evaluate the research on its quality, significance of the work, and the clarity of the informal discussion. Three semi-finalists chosen from this round of competition will present their research during a conference session. The Student Research Competition winners will be recognized prior to the keynote speaker, Bill Gates, on the last day of the conference.

## STUDENT VOLUNTEERS

**Chair: Maria Elena (Helen) Chávez Echeagaray,**  
ITESM, Campus Guadalajara, [sv@oopsla.acm.org](mailto:sv@oopsla.acm.org)



The Student Volunteer program is an opportunity for students from around the world to associate with the top people in object-oriented technology, research, and software development. In return for about ten hours of their time, student volunteers receive complimentary registration, free admission to tutorials (space permitting), and other benefits. A limited number of volunteers may receive some financial support in return for additional volunteer hours. In past years, job assignments included assisting with symposia and

panels, checking badges at doors, manning the information booth, helping with traffic flow and general *go-for* assistance to keep the conference running smoothly.

Student volunteers need to be enrolled in a full-time undergraduate or graduate program at the time of the conference. For more details, or to submit an application (before **19 July 2002**), visit the Student Volunteer website <http://oopsla.acm.org/~sv>.





## SPECIAL EVENTS

OOPSLA 2002 provides the opportunity to mingle with the recognized leaders in object technology in a more casual setting through a number of social activities designed just for that purpose. In addition, the OOPSLA Courtyard, in the center of the Exhibit Hall, will offer food and beverage concessions, Poster Session, the OOPSLA CodeFest, Camp Small Talk and one-on-one interaction with Conference speakers and authors.

Name badges will be required to attend events.

**Monday 17:30-19:30**

### Tutorial and Workshop Reception

The reception is open to anyone who attends (or leads) at least one Tutorial or Workshop. This is an excellent forum to share information on the day's activities.

**Tuesday 17:30-19:30**

### OOPSLA 2002 Welcome Reception

All full (and Tuesday one-day) conference registrants are invited to attend the Welcome Reception. OOPSLA attendees are encouraged to view the OOPSLA 2002 Poster Session during the Welcome Reception.

**Tuesday 19:30-20:00**

### Newcomers Orientation

A special session has been arranged for OOPSLA newcomers. If you have never been to an OOPSLA before, this is a unique opportunity to meet other first-time OOPSLA participants early in the conference and to learn about the facets of OOPSLA and how to maximize your OOPSLA experience.

**Wednesday 17:00-18:30**

### Exhibitors Reception

Located in the Exhibit Hall, this reception is open to all full conference registrants. Stop by for chance to view the exhibits and visit Geek Alley.

**Thursday 19:00-23:00**

### Evening at "Experience Music Project"

Ride the Monorail to this museum and you are in for something very different! The museum was designed by world-renowned architect Frank O. Gehry and is located at the base of the Space Needle. Both the building's architecture and its unparalleled collection of music-related artifacts have made EMP a very popular destination for locals and visitors alike.

- Crossroads, the main exhibit area, combines rare artifacts with imagery and multimedia to recount stores that span a century of American popular music. The exhibit pays tribute to many Northwest artists and other musicians from around the country, the inspiring artistry of Jimi Hendrix, the development of the electric guitar and more. Musical revolutions from jazz and blues to the birth of rock n' roll and the hip-hop explosion are explored, providing fresh insight into the diverse expressions of musical creativity.
- The Sound Lab will be open for those who enjoy playing, performing and moving to music. Whether a novice or professional musician, you can explore your creative potential as you perform to a virtual concert audience or engage with interactive instruments.
- The Sky Church will open its doors for OOPSLA guests who want to dance. Light refreshments will be available. For a sneak peak of EMP, visit <http://www.emplive.com>.

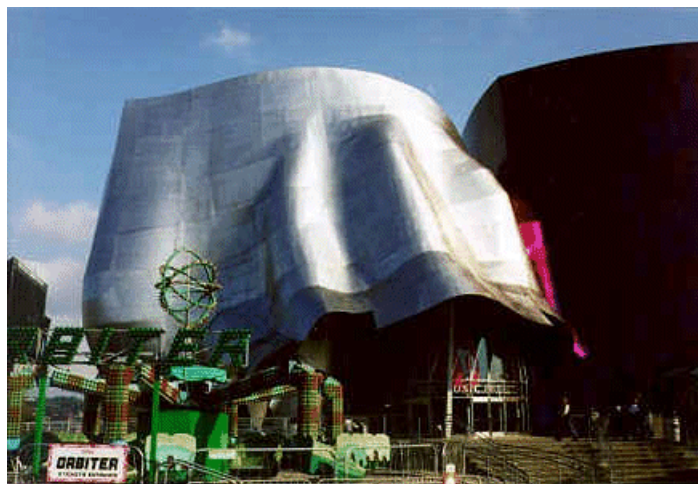


Photo by: Russell Reising, University of Toledo

**Friday 17:00-18:00**

### OOPSLA 2003 Kick-off (Ice Cream Social)

The Ice Cream social serves as the kick-off event of next year's OOPSLA where you will be able to pick up your OOPSLA 2003 poster. It is also the setting for the DesignFest/CodeFest Wrap-up. Throughout the week, teams of software designers have taken part in the DesignFest sessions. In addition, a small number of student teams (known as CodeFest) have implemented some of these designs. The Ice Cream Social is a chance for OOPSLA participants, whether or not they participated in DesignFest, to see the designs produced during the earlier sessions, as well as to view demos of the final software. Designers will be available to discuss their experiences (both good and bad) of working in their design team, and the CodeFest teams will describe the problems they encountered while implementing the designs.



## BIRDS OF A FEATHER SESSIONS (BOFS)

Birds of a Feather Sessions are informal gatherings that provide a forum for discussion of a particular topic, tool, product, etc. Several rooms will be available for BOFs each evening from Monday through Thursday. Rooms may be reserved by any Conference attendee on a first-come, first-served basis by signing up at the Information Booth at the Conference. There will be no advance reservations for these rooms prior to the Conference.

BOF organizers are responsible for providing descriptive information for their session, which OOPSLA will post on the BOF notice board. BOFs require no registration or submission for attendance. All OOPSLA attendees may attend any BOF session(s).

## CAMP SMALLTALK

Chair: Ralph Johnson, *University of Illinois Urbana Champaign*,  
[johnson@cs.uiuc.edu](mailto:johnson@cs.uiuc.edu)

Camp Smalltalk combines the power of Smalltalk with the energy of open source. We will set up Camp in the exhibition hall. Stop by if you want to join in, to see Smalltalk in action, or just to meet an old friend. For further information, see the OOPSLA website at <http://oopsla.acm.org>.



# EXHIBITS

## Chair: Jim Johnson, Meetings and More



Attended by more conference registrants than any other event, the OOPSLA Exhibits are where OO software developers, publishers, technology researchers, framework architects, training consultants and recruiters go for a “meeting of the minds” and an introduction to the latest object-oriented offerings.

This is the one place you can reach potential customers with news and demonstrations of your latest products, publications or services.

The OOPSLA 2002 Exhibits will be held at the Washington State Convention and Trade Center (WSCTC), also the site of other Conference activities and programs.

Only at OOPSLA can exhibitors benefit from the following features:

- Double-sized (10x20) booth spaces at discount rental rates with ample area for product display and visitors. Larger spaces are also available at discount rates.
- A Bartizan lead retrieval system, for enhanced ROI and increased sales- at no extra cost!
- A complimentary Conference registration for each 100 sq. ft. of booth space rented. Conference registrations allow exhibitors to interact with other attendees at selected technical sessions and social functions. Each Conference registration is valued at \$300-500!
- The “OOPSLA Courtyard” on the Exhibits floor will provide food & beverage concessions; message boards; the OOPSLA “CodeFest”; “Camp Smalltalk”; and the popular “Meet the Speakers” lounge.
- Conference refreshment breaks will take place in the Exhibits.
- The popular Publishers’ Pavilion, where attendees can browse for the latest periodicals and books.
- Geek Alley, where the latest technology in personal computing devices and peripherals can be purchased directly from the exhibiting company.
- OOPSLA Conference Posters are integrated with exhibit booths.
- OOPSLA 2002 exhibitors can link the OOPSLA web site to their company page to let the object technology community know where to find you in Seattle.
- The Exhibitors’ Reception, open to all Conference participants, will take place on the Exhibits floor.

## Exhibit Hall

Exhibit Hall A of the WSCTC will be the exclusive domain for the 2002 OOPSLA Exhibits. Hall A is close to OOPSLA Registration and other primary Conference events. OOPSLA Conference Demos will take place in the Exhibit Hall.

## Preliminary Exhibits Agenda

Note: Agenda subject to change

### Tuesday, 5 November 2002

8:00 am – 20:00 pm Exhibitor Setup

### Wednesday, 6 November 2002

10:00 am – 13:30 pm Exhibits Open

(13:30-15:00 pm) Exhibits Closed

15:00 pm – 18:30 pm Exhibits Open

17:00 pm – 18:30 pm Exhibitors’ Reception

### Thursday, 7 November 2002

10:00 am – 17:00 pm Exhibits Open

### Friday, 8 November 2002

10:00 am – 14:00 pm Exhibits Open

## For Booth Rental Information

Exhibitor Prospectus also available at: [www.oopsla.acm.org](http://www.oopsla.acm.org)

Jim Johnson

Meetings & More Inc.

14449 N. 73<sup>rd</sup> St.

Scottsdale, Arizona 85260 USA

Voice: 1-480-998-3992

Fax: 1-480-998-7838

Email: [meetmore@aol.com](mailto:meetmore@aol.com)

## 2001 OOPSLA Exhibitors

101 Communication

ACM

ACM SIGAda

Addison-Wesley

Cambridge University Press

Cap Gemini – Ernst & Young

Cincom Systems

ETS-GRE Computer Science

Fresher Information Corp

IBM

Institute of Software Architects

Jones and Bartlett Publishers

McGraw-Hill Higher Education

MetaCase Consulting

Microsoft Corporation

Morgan Kaufmann

Object Management Group

Object Mentor Inc

Precision System Design

Prentice Hall PTR

Progressive Insurance

Prophecy Americas Inc

Roaming Media Inc

Rogue Wave Software

Seattle CVB

Springer-Verlag New York

Sysnetics Inc

Tampa Bay Computers

TogetherSoft Corporation

John Wiley & Sons

**OOPSLA 2002 Exhibits Booth Application & Contract**  
**5-8 November 2002 • Washington State Convention & Trade Ctr • Seattle, WA USA**

## Instructions

Please read this contract carefully. Print or type all information requested. Complete all sections of the contract.

### *Pay by Check*

Make Checks Payable To:  
 OOPSLA 2002 (U.S. funds only).

Send the completed original contract with appropriate deposit to:

OOPSLA 2002 Exhibits  
 Meetings & More Inc.  
 14449 N. 73rd St.  
 Scottsdale AZ 85260 USA  
 Phone: 1-480-998-3992  
 Fax: 1-480-998-7838

### *Pay by Credit Card*

To pay for booth rental with a credit card, please provide an authorization letter on your company stationery stating the booth rental rate, credit card type, credit card number, expiration date, cardholder's name (printed) and cardholder's signature. This authorization may be faxed or mailed.

Confirmation of your booth assignment will be sent to you by mail.

We hereby submit our application to exhibit at OOPSLA 2002. We acknowledge that we are bound by all regulations set forth in the ACM/OOPSLA terms, conditions and rules for exhibiting. Enclosed is our payment for booth space. If we cancel our booth space, such cancellation shall be considered in default and any monies paid may be retained by OOPSLA as liquidated damages. We pledge that we will not sponsor or participate in any private group functions or operate a hospitality suite during OOPSLA Conference or Exhibit hours without the expressed, written consent of OOPSLA authorities.

Authorized Signature \_\_\_\_\_

Name (please print) \_\_\_\_\_

## Please Type or Print Clearly

Company Name: _____	Email: _____
Mailing Address: _____	City: _____ State: _____
Zip Code: _____	Country: _____
Phone: _____	Fax: _____
Contact person (all exhibits-related material sent to this person): _____	
Address of Key Contact (if different from above): _____	

## Booth Rental Rates

Full payment required with submission of Application & Contract.  
 Note: 10 feet x 20 feet is smallest booth size. Rental includes (2) two Conference registrations and Bartizan lead retrieval system. Each additional 10 x 10 booth rental includes (1) Conference registration.

10x20 Booth = \$2500 (First-time OOPSLA exhibitors: \$2000)  
 10x30 Booth = \$3000 (First-time OOPSLA exhibitors: \$2500)  
 20x20 Booth = \$3500 (First-time OOPSLA exhibitors: \$3000)  
 20x30 Booth = \$4500 (First-time OOPSLA exhibitors: \$4000)

Size of Booth Required:    10x20   10x30   20x20   20x30

At its discretion, ACM reserves the right to relocate exhibitors or re-design the floor plan to improve the overall effectiveness of the Exhibits.

We prefer not to be located next to the following (2) competitors:

- 1 \_\_\_\_\_
- 2 \_\_\_\_\_

We will feature the following products/services in our booth:

\_\_\_\_\_

\_\_\_\_\_

## For OOPSLA 2002 Use Only

Contract Received: _____	
Assigned Booth #: _____	
Deposit Rec'd: _____	Balance Due: _____
Total Rental: _____	Conf. Date: _____
Pd in Full: _____	Seniority #: _____

# HOUSING INFORMATION

## Downtown Seattle Housing Information

We are using an Internet Housing Bureau to process reservations for this conference. To make your hotel reservations, go to the web address noted on the form and follow the online instructions. This is a secure site. If you do not have Internet access, you may either fax, phone or mail your reservation request to the number or mailing address noted below. We encourage you to utilize the Internet option for reserving your registration. Online reservations will receive an immediate confirmation of their reservation going directly to their email account. Online reservation will also allow you to access your reservation at any time before cut-off-date of **4 October 2002** to either make changes to or cancel your reservation.

### ① Sheraton Seattle Hotel

1400 Sixth Ave.

Seattle, WA 98101

Room Rates:      \$157.00 per night Single  
                          \$177.00 per night Double

The Sheraton Seattle Hotel & Towers is located in the heart of downtown Seattle, one block from the Washington State Convention & Trade Center. Experience the unparalleled service of the Sheraton, including a 35<sup>th</sup> floor pool, Jacuzzi and health club and newly-renovated rooms.

There are two restaurants and a lounge available along with 24 hour room service.

### ② WestCoast Grand Hotel

1415 Fifth Ave

Seattle, WA 98101

**Room Rates:**      \$126 single (\$10 each additional person)

WestCoast Grand Hotel offers 300 deluxe rooms with superb views of Elliott Bay and the city. Experience indoor and outdoor dining on the largest patio in downtown in the Terrace Garden Restaurant and Lounge-or the Elephant & Castle Pub for the best of British fare. Wireless high-speed Internet available in guest accommodations and meeting space. Located in the heart of downtown Seattle only two blocks from the Washington State Convention & Trade Convention & Trade Center. Let WestCoast hospitality take care of your meeting and travel needs

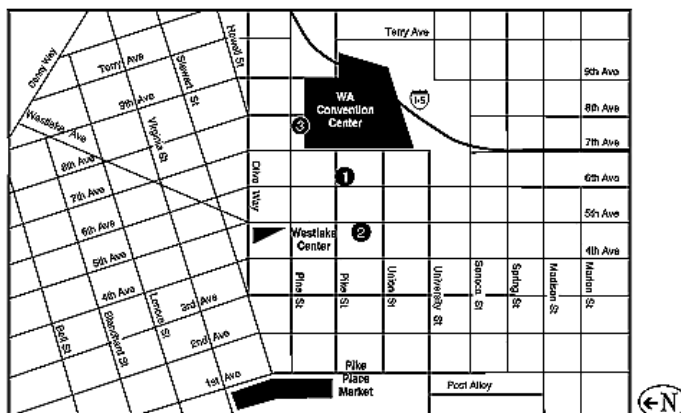
### ③ The Roosevelt Hotel

1531 Seventh Ave.

Seattle, WA 98101

Room Rates:      \$128 single and double (\$10 each additional person)

Located one-half block from the Convention Center in the heart of downtown. The Roosevelt has been traditionally restored and is elegant and warm. The hotel is completely modern in amenities and guest services.



## Deposit Information

All hotels require a credit card guarantee of one night's hotel room & tax with each reservation request. Requests received without a deposit will be returned and will not be processed. Please fill out the credit card information entirely. Credit Cards must be valid through November 2002 in order to be considered a proper deposit.

## Room Rates/taxes

To take advantage of the special OOPSLA rates, be sure to book your reservation by 12:00 noon (PST) on 4 October 2002. After this date the official OOPSLA blocks will be released the hotels may charge significantly higher rates. All rates are per room per night and are subject to an 15.8% tax (subject to change). Hotels may charge additional fees for rooms with more than two occupants. When making a reservation, please provide room and bedding preferences in the Special Needs section on the reverse side of this form. The hotels will assign specific room types upon check in, based upon availability. Please be advised that requests are not guaranteed.

## Modifications/cancellations

Reservations secured by credit card may be changed or canceled through the Housing Bureau until 21 October 2002. Cancellation received on or after 4 October 2002 through 1 November 2002 will be charged a \$25 processing fee. On or after 2 November 2002, cancellations will assess a forfeiture of one night's room and tax.

## Reservation Confirmations

Confirmation will be sent after each reservation booking, modification and/or cancellation. Review it carefully for accuracy. If you do not receive a confirmation via e-mail, fax, phone, or mail within 14 days after any transaction, please contact the Housing Bureau by phone or email at [hotelres@seeseattle.org](mailto:hotelres@seeseattle.org).

## Questions? Concerns?

Questions may be e-mailed to:  
[hotelres@seeseattle.org](mailto:hotelres@seeseattle.org),  
 or call 1-888-877-0255, 1-206-461-5881  
 Monday-Friday from 8:30am-17:00pm PST.

# OOPSLA 2002

## 4 - 8 November 2002 — Seattle, Washington

### Instructions

**Reservations can be made between 1 July 2002 – 4 October 2002**, by choosing **one** of the following methods:

- **Internet**  
Book your reservation online using the interactive site at [www.acm.org/sigplan/oopsla/oopsla02](http://www.acm.org/sigplan/oopsla/oopsla02)
- **Phone**  
Call the OOPSLA Housing Bureau at 1-888-877-0255 or for International callers 1-206-461-5881.
- **Fax**  
Send a completed form, one copy per room to 1-206-461-5853.
- **Mail**  
Send a completed form, one copy per room to:  
OOPSLA 2002 Housing  
520 Pike Street, Suite 1300  
Seattle, WA 98101 USA

### Room Rates

To take advantage of the special OOPSLA rates, be sure to book your reservation by 12:00 noon (PST) on 4 October 2002. After this date the hotels may charge higher rates.

### Modifications & Cancellations

Reservations secured by credit card may be changed or canceled via the web or through the Housing Bureau until **21 October 2002**. Cancellations received on or after **4 October 2002** will be assessed a \$25 processing fee. Do not contact the hotel directly until after **21 October 2002**. On or after **2 November 2002**, cancellations will assess a forfeiture of one night's room and tax.



### SEND CONFIRMATION TO: (fill this portion out completely)

First: \_\_\_\_\_  
 Company: \_\_\_\_\_  
 Address: \_\_\_\_\_  
 Zip Code: \_\_\_\_\_  
 Phone: \_\_\_\_\_

Last: \_\_\_\_\_  
 Email: \_\_\_\_\_  
 City: \_\_\_\_\_ State: \_\_\_\_\_  
 Country: \_\_\_\_\_  
 Fax: \_\_\_\_\_

### Hotel Information Form

Arrival Date: \_\_\_\_\_

Departure Date: \_\_\_\_\_

- ☐ **Sheraton Seattle** (OOPSLA Headquarters Hotel)  
☐ WestCoast Grand Hotel  
☐ The Roosevelt Hotel

#### Hotel Selection:

First Choice: \_\_\_\_\_

Second Choice: \_\_\_\_\_

Third Choice: \_\_\_\_\_

Reservations will be processed on a first come, first serve basis.

#### Room Type: (must fill out both)

Number of people in room: \_\_\_\_\_

Number of beds in room: \_\_\_\_\_

List all occupants in room: \_\_\_\_\_

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Special Needs: \_\_\_\_\_

\_\_\_\_\_  
 \_\_\_\_\_

#### Please check one:

- ☐ Smoking  
☐ Non-smoking

### Deposit Information

HOTELS REQUIRE A CREDIT CARD GUARANTEE OF ONE NIGHT'S ROOM & TAX (CURRENTLY 15.8%) WITH EACH RESERVATION REQUEST. HOUSING FORMS RECEIVED WITHOUT A VALID CREDIT CARD WILL BE RETURNED AND WILL NOT BE PROCESSED.

Account Number: \_\_\_\_\_

Exp. Date: \_\_\_\_\_

Card Type: American Express, MasterCard, Visa, Discover, Diners

Name of Cardholder: \_\_\_\_\_

Cardholder's Signature: \_\_\_\_\_



# HOUSING SCHOLARSHIP FOR QUALIFIED ACADEMIC FACULTY

**Chair: Helen Sharp, The Open University, UK,  
[h.c.sharp@open.ac.uk](mailto:h.c.sharp@open.ac.uk)**

OOPSLA 2002 is proud to announce again this year the availability of a limited number of housing scholarships for qualified academic faculty. These scholarships are sponsored by the Association of Computing Machinery (ACM) Special Interest Group on Programming Languages (SIGPLAN). Scholarship recipients will, in effect, receive half-price rates at the main conference hotels. OOPSLA will reimburse you for half of your hotel room charges, for a stay of at least two and at most six nights.

Further information is available from the OOPSLA 2002 website <http://oopsla.acm.org>. Questions should be directed to the coordinators of this program at [Academic\\_Housing@oopsla.acm.org](mailto:Academic_Housing@oopsla.acm.org).

Only faculty from educational institutions are eligible to apply.



# REGISTRATION INFORMATION

**Last day for reduced rates: Thursday, 26 September 2002**

**Last day to advance register: Thursday, 17 October 2002**

To qualify for reduced rates, registration forms must be sent by 26 September 2002 – that is, sent online, faxed, or postmarked by that date. The last date to advance register by fax, mail, or online is 17 October 2002. After that, you can register only at the Conference.

Registration forms must be accompanied by full payment in order to be processed. Incorrect credit card numbers and declined credit are considered no payment and registration will not occur.

In past years, OOPSLA Tutorials have sold out very early. It is strongly recommended that you register well in advance of the 26 September 2002 deadline. Heavy fax traffic is expected on that day, and OOPSLA cannot be responsible for forms not received due to busy phone lines.

Online registration is available at <http://oopsla.acm.org>.

No telephone registrations will be accepted.

## Payment information

Payment in U. S. dollars may be made by personal or company check, money order, or credit card.

- Mailed checks must be payable to OOPSLA 2002.
- Faxed and online registrations must be paid by credit card.
- Purchase orders, government vouchers, wire transfers, telephone or electronic mail registrations will not be accepted.
- International payment must be by credit card, International Money Order, or drafts drawn against a U.S. bank account.

## Refunds/Transfers

Cancellations made in writing and faxed or postmarked by 17 October 2002 will be accepted subject to a \$50 cancellation fee. Refunds will be made within four weeks of the end of the conference. Cancellations will not be accepted after 17 October 2002. “No-shows” are not refundable and are liable for the full registration fee. Instead of canceling, your registration may be transferred by giving a colleague a written authorization.

Conference materials (Proceedings, Tutorial Notes, etc.) which are not picked up at the conference, will not be shipped to the registrant after the conference.

## What your Registration Includes

- Conference registration includes admission into Technical Sessions, Exhibits, Welcome Reception, Exhibits Reception, one Special Event ticket, and the Proceedings.
- Tutorial registration includes admission into the selected Tutorial, one copy of the Tutorial Notes for that session, Lunch for the day of the selected Tutorial, Tutorials Reception, and Exhibits. You do not have to register for the conference in order to attend Tutorials.
- The Educators’ Symposium fee includes admission into the Symposium, the Educators’ Symposium notes, Lunch on the day of the Symposium, and Exhibits.
- Workshop participants must be registered for the conference or for the day of the workshop, and are invited to the Workshop Reception.
- Student conference registration includes admission into Technical Sessions, Tutorials, Exhibits, Welcome Reception, Exhibits Reception, and the Proceedings. Student fees do not include the Special Event ticket.

## Important New Policy regarding Tutorial Registration for Students

- Students may attend tutorials of their choice free of charge, and tutorial notes will be available for sale in the registration area. Seating availability will be determined first-come first-served outside the tutorial rooms, ten minutes before the start of each tutorial. Please note that seating priority will be reserved to confirmed Student Volunteers, and tutorial lunches are not included in the student fees.
- Students wishing to guarantee seating in tutorials may purchase seats at the regular rates published for ACM members and non-members.
- Panel participants, Session Chairs, and Speakers must register and pay for the conference.
- Exhibits-only passes will be available onsite at no charge.

## Confirmation

Confirmation letters will be sent by mail. Please allow up to two weeks for mailed confirmation of your registration. We cannot fax confirmations.

## Sell-out Policy

All registration will be processed first-come, first-served, based on date of receipt. On-site registration will be accepted subject to the availability of space and materials. OOPSLA reserves the right to limit Conference, Tutorials, Workshops, Educators’ Symposium, DesignFest®, and Special Event registrations.

## Registration Form Instructions

### Section 1:

Carefully print or type all information in this section to avoid spelling errors on your badge.

### Section 2:

- Member fees apply to ACM, SIGPLAN, or SIGSOFT members.
- Students must submit a letter from the student’s department confirming academic status as a full-time student at the time of registration.

### Section 3:

- Circle the chosen Tutorial numbers and add up the units for each day.
- You may take a maximum of two Tutorial units per day. Avoid scheduling conflicts.
- Participants in the Educators’ Symposium should not schedule any Tuesday Tutorial.
- Enter the total number of Tutorial units in section 5 for Tutorial fee computation.
- Every effort will be made to assign your first choice Tutorial. An alternate Tutorial will be assigned only if you have provided a prioritized list of alternative Tutorials at the bottom of section 3.
- Tutorial availability information can be found at the OOPSLA 2002 Web site or <http://www.regmaster.com/oopsla2002.tutavail.html>.

### Section 4:

Please circle all appropriate fees for Conference, Tutorials, and Educators’ Symposium.

**Section 5:**

- Fill in all fees as appropriate, referring to the table of fees in section 4.
- A full set of Tutorial Notes is available, and cost includes shipping. Notes will be shipped after the conference directly from the printer to the registrant's address, unless otherwise notified. Tutorial Notes will not be sold after the conference.
- If paying by VISA, MasterCard, American Express or Diner's Club, please provide the information requested in full to avoid processing delays, including the credit card's billing address if it differs from the attendee's address in Section 1.

**Registration Addresses and Fax Number****Online registration:**

<http://oopsla.acm.org>

**Mail:**

OOPSLA 2002  
c/o Registration Systems Lab  
61 Alafaya Woods Blvd.  
MS #199  
Oviedo, FL 32765 USA

**Fax:** 1-407-366-4138

The registration addresses above are for conference registration only. Hotel reservation forms must be sent to the address provided in the Housing section.

For **registration information** only:

Phone: 1-407-971-4451

E-mail: [registration@oopsla.acm.org](mailto:registration@oopsla.acm.org)



What Your Registration Includes

What Your Registration Includes:

OOPSLA '02	Technical Program	Tutorials	Tutorial Notes	Educators' Symposium	Workshops	Lunch	Exhibits	Proceedings	Tutorials/Workshops Reception	Welcome Reception	Exhibits Reception	"Experience Music Project"
Conference	•				•		•	•	•	•	•	•
Tutorial registration		•	•			(tutorial day)	•		•			
Educators' Symposium				•		(Tuesday)	•		•			
Student registration	•	•					•	•		•		•
Exhibits only							•					

• Workshops are by invitation only, and accepted participants must register for the conference or for the day of the workshop.

# Advance Registration Form

**1** First name \_\_\_\_\_ M.I. \_\_\_\_\_ Last name \_\_\_\_\_

Company or School \_\_\_\_\_

Address \_\_\_\_\_

**2 AFFILIATION**

☐ ACM Student # \_\_\_\_\_

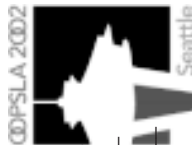
☐ Student school ID # \_\_\_\_\_

☐ ACM | SIGPLAN | SIGSOFT # \_\_\_\_\_

☐ Non-Member

Mail this form to:

OOPSLA 2002  
61 Alafaya Woods Blv  
PMB 199  
Oviedo, FL 32765 US  
Fax: +1-407-366-413



City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_ Country \_\_\_\_\_

Phone \_\_\_\_\_ Fax \_\_\_\_\_

E-mail: \_\_\_\_\_

☐ I want to be included on the attendee's mailing list ☒ I want to receive future OOPSLA announcements b.

☐ I want my e-mail on the attendee's mailing list ☐ I want to receive future OOPSLA announcements b.

☐ I want to be included on the exhibitor's mailing list ☐ I have special needs (please specify): \_\_\_\_\_

☐ I am staying at a conference (Hotel, West Coast Grand Hotel, Renaissance Hotel, etc.) NOT use the Housing bureau.

**3 TUTORIALS** (Please circle Tutorial numbers below)

	UNITS									
Monday Full-day	2	1	2	3	4	5	6			
Monday Morning	1	7	8	9	10	11				
Monday Afternoon	1	12	13	14	15	16	17			
Tuesday Full-day	2	18	19	20	21					
Tuesday Morning	1	22	23	24	25	26	27	28	29	
Tuesday Afternoon	1	30	31	32	33	34	35	36	37	38
Wednesday Full-day	2	39	40	41						
Wednesday Morning	1	42	43	44						
Wednesday Afternoon	1	45	46	47	48					
Thursday Afternoon	1	49	50	51	52	53	54			

Total Units: \_\_\_\_\_ Add tutorial units above. The maximum number of units is 7.

2nd Choice: Monday: \_\_\_\_\_ Tuesday: \_\_\_\_\_ Wednesday: \_\_\_\_\_ Thursday: \_\_\_\_\_

**4 CONFERENCE AND TUTORIAL FEES in U. S. DOLLARS**

(circle the appropriate fees)	On or Before September 26, 2002		After September 26, 2002	
	ACM Student	Member	Non-member Student	Non-member
Conference Fee	\$80	\$100 \$365	\$445	\$80 \$100 \$465
One-day Fee (Mon. or Tue.)	\$40	50	185	225
Tutorials: per Unit Fee	260	290	260	290
Educators' Symposium	30	40	120	140

\* Students may attend tutorials at no charge on a space available basis. Please see details online at [oopsla.org](http://oopsla.org)

**5 PAYMENT COMPUTATION**

Conference Fee (Wednesday, Thursday, Friday) \$ \_\_\_\_\_

One-day Fee: ☐ Monday ☐ Tuesday \$ \_\_\_\_\_

Tutorials Fee: \_\_\_\_\_ unit(s) x \$ \_\_\_\_\_ = \$ \_\_\_\_\_

Educators' Symposium \$ \_\_\_\_\_

Additional Experience Music Project: ticket(s) x \$70 = \$ \_\_\_\_\_

Full set of Tutorial Notes: \_\_\_\_\_ set(s) x \$70 = \$ \_\_\_\_\_

TOTAL FEES ENCLOSED \$ \_\_\_\_\_

**6 DESIGNFEST® SIGN UP**

Forms will NOT be processed without payment checks payable to OOPSLA 2002. If paying by VISA, MasterCard, American Express, or Diner's Club, please complete the following section in full.

Card No: \_\_\_\_\_ Expires: \_\_\_\_\_

Cardholder's name \_\_\_\_\_

Cardholder's signature \_\_\_\_\_

Credit Card billing address if different from section 1 \_\_\_\_\_

Where did you hear about OOPSLA

☐ a. ACM Publications

☐ b. IEEE Computer

☐ c. ADT

☐ d. Google - UseNet

☐ e. Java & OO User Groups/SIGs

☐ f. E-mail via ACM Distribution

☐ g. Internet Ad via ZDNet/CNet

☐ h. Microsoft Website

☐ i. Friends & Associates

☐ k. Advance Program

Please check all that apply:

☐ User ☐ Academic ☐ Developer ☐ Executive

Space is limited. There is no charge for participation, but you must register for the full conference. The DesignFest is highly interactive! Passive observers should not sign up. Please supply your e-mail address in section 1 above. This is my first OOPSLA



# TRANSPORTATION INFORMATION

## Airlines

We have arranged special discounts with American Airlines. To take advantage of these special discounts please call the American Airlines Meeting Services Desk at **1-800-433-1790** or your local travel agency and refer to authorization number **A04N2AI**. As an official carrier, American Airlines has provided us with the following discounts:

- 10% off the full published fares
- 5% off the lowest fares

[http://www.acm.org/sig\\_volunteer\\_info/american.html](http://www.acm.org/sig_volunteer_info/american.html)

## SeaTac International Airport

Located 40 minutes from downtown Seattle, the SeaTac International Airport is a world-class facility that consistently ranks as one of the world's top airports. It is served by 24 major and regional carriers, including:

- |                         |                |
|-------------------------|----------------|
| • Alaska Airlines       | 1-800-682-2221 |
| • Continental Airlines  | 1-800-525-0280 |
| • Frontier Airlines     | 1-800-432-1359 |
| • Scandinavian Airlines | 1-800-221-2350 |
| • Southwest Airlines    | 1-425-450-0990 |
| • Northwest Airlines    | 1-800-225-2525 |
| • United Airlines       | 1-800-241-6522 |

## Taxis and Shuttles - Airport and Downtown

Average taxi service from SeaTac International Airport to downtown is \$35.00 one way. Transportation from the airport is provided by a shuttle service call "Gray Line of Seattle". They suggested you make reservations by calling 1-800-426-7505, or check out their web site at <http://www.graylineofseattle.com> (cost: \$8.50-one way).

- |                         |                                 |
|-------------------------|---------------------------------|
| • Sheraton Seattle      | \$8.50 one way, \$14 round trip |
| • WestCoast Grand Hotel | \$8.50 one way, \$14 round trip |
| • The Roosevelt Hotel   | \$8.50 one way, \$14 round trip |

## Amtrak, Greyhound and Car Rental

Seattle is served by Amtrak National Rail Passenger Service (1-800-USA-RAIL) and Greyhound Bus Lines (206) 626-6088.

Car rental is available at the airport at the baggage claim level, companies represented include:

- |                     |                                  |
|---------------------|----------------------------------|
| • Alamo Rent A Car  | 1-206-433-0182                   |
| • Budget Rent-A-Car | 1-800-527-0700 or 1-206-682-CARS |
| • Dollar Rent-A-Car | 1-800-800-4000 or 1-206-433-6768 |

# ACM / SIGPLAN / SIGSOFT



ACM, the Association for Computing Machinery, founded in 1947, is the world's first educational and scientific computing society. Today, its members—over 75,000 computing professionals and students world-wide—and the public turn to ACM for authoritative publications, pioneering conferences, and

visionary leadership for the new millennium. ACM offers its members a vast array of IT information resources, including the ACM Portal, which includes over 750,000 pages of text, the Online Guide to Computing Literature (with 350,000 bibliographic citations extending far beyond ACM's own literature), the Online Computing Reviews Service, customized personal services, and more.

For a complete listing of ACM's Professional and Student membership benefits and options, visit <http://www.acm.org/joinacm>.

ACM SIGPLAN explores programming language concepts and tools, focusing on design, implementation, and efficient use. Its members are programming language users, developers, implementers, theoreticians, researchers, and educators. The monthly newsletter "ACM SIGPLAN Notices" publishes several conference proceedings issues, regular columns and technical correspondence. This SIG offers an additional newsletter (FORTRAN Forum), on a subscription only basis. SIGPLAN sponsors four major annual conferences: the OOPSLA conference in object-oriented technology, the Conference on Programming Language Design and Implementation (PLDI), the major professional conferences in the field; the Symposium on Principle of Programming Languages (POPL), and the International Conference on Functional Programming (ICFP), held periodically in Europe.

## Contact ACM to join.

### Voice:

1-800-342-6626: US & Canada

1-212-626-0500: NY & Global

### Fax:

212-944-1318

### E-mail:

[acmhelp@acm.org](mailto:acmhelp@acm.org)

### Web:

<http://www.acm.org/>

<http://www.acm.org/sigplan/>

<http://www.acm.org/sigsoft/>

## SEATTLE FACTS

Ask any Seattleite where to get the best cup of espresso, and you'll receive an enthusiastic answer. Ask if it's true that it rains less in Seattle than it does in New York, Atlanta, Boston and even Miami, and you may receive an evasive reply. That's because it's true, and Seattleites are very protective of their little piece of paradise. After all, it's well known that the bluest skies you've ever seen are in Seattle, not somewhere down in the southern California. Seattle is a lovely place to come for a convention or meeting – if you have the discipline to stay indoors.

A scenic blend of mountains and the subtle hues of evergreens and flowered hillsides, the city are virtually ringed by water with views of the Olympic and Cascade Mountains and the indomitable 14,410-foot Mt. Rainier. Ferries cross the Puget Sound to Bainbridge and Vashon Islands and beyond to the San Juan's.

Because two mountain ranges flank the city, the climate is temperate year-round, and gardens thrive even in the mid-winter. Rain is usually a gentle mist that freshens the air and keeps the landscape naturally green and lush.

A booming waterfront is testament to Seattle's Maritime connection. Attractions include Pike Place Market, the nation's oldest working farmers market since 1907; Odyssey, the Maritime Discovery Center; The Seattle Aquarium; harbor tours; and simply strolling amidst fresh salt breezes and smells of crab and cracked sourdough bread, with views of ferry and freighter traffic on Elliott Bay and the Olympic Mountains.

Weather Temperature for November: High 51° - Low 41°

### Amusements, Attractions & Activities

#### Gameworks

1511 7<sup>th</sup> Ave  
Seattle, WA 98101  
Phone: 1-206-521-0952

Gameworks is a unique place where people can connect and play the hottest games available.

#### Pacific Science Center

2000 Second Ave N  
Seattle, WA 98109  
Phone: 1-206-443-2001

At Pacific Science Center, explore the Tropical Butterfly House, Tech Zone, robotic dinosaurs and more. The Boeing IMAX Theater offers thrilling large-format films, including 3D.

#### The Seattle Aquarium

Pier 59  
1483 Alaskan Way  
Seattle, WA 98109  
Phone: 1-206-386-4300

Award-winning exhibits of aquatic life, including a Pacific Coral Reef, Underwater Dome, working salmon ladder, and naturalistic Puget Sound habitats. Highlights include the Discovery Lab, the new watershed exhibits, sea otters and other marine mammals.

#### Seattle Center

305 Harrison  
Seattle, WA 98109  
Phone: 1-206-684-7200

74 acres of arts, entertainment, sport & recreation, shopping, dining, educational and cultural adventures await you at the nation's preeminent gathering place.

#### Space Needle

219 Fourth Ave. N  
Seattle, WA 98109  
Phone: 1-206-443-2111

Revolving restaurant. Observation Deck, banquet level and gift shops. Elevator ride complimentary when dining.

#### Underground Tours

608 First Ave.  
Seattle, WA 98104  
Phone: 1-206-684-4646

Seattle's hidden treasure in Pioneer Square. Historic, tongue-in-cheek, narrated walking tour. The tour begins in a restored 1890's saloon, with a 20-minute introduction. Walks last 1 hour.

#### Woodland Park Zoo

5500 Phinney Ave. N  
Seattle, WA 98103  
Phone: 1-206-684-4800

Reconnect with nature at this top award-winning zoo and see why it's hailed as among the finest in the world.

#### Chateau Ste. Michelle

14111 NE 145<sup>th</sup>  
Woodinville, WA 98072  
Phone: 1-425-415-3300

Washington's pioneer producer of classic European wines, 25 minutes NE of Seattle.

#### Columbia Winery

14030 NE 145<sup>th</sup>  
Woodinville, WA 98072  
Phone: 1-206-488-2776

Complimentary wines tastings offered daily.

#### Pyramid Brewery & Alehouse

1201 1<sup>st</sup> Ave. S  
Seattle, WA 98134  
Phone: 1-206-68-BEERS

Tours daily. Sample Pyramid Ales and Thomas Kemper Lagers & Sodas while enjoying hearty "pub grub"

#### Redhook Ale Brewery

3400 Phinney Ave. N  
Seattle, WA 98103  
Phone: 1-206-548-8000

Tours available daily at Woodinville facility.

#### Experience Music Project

325 Fifth Ave. N  
Seattle, WA 98109  
Phone: 1-206-EMP-LIVE

Explore American popular music at EMP through interactive exhibits, rare artifacts and one-of-a-kind ride at Seattle Center

#### Act-A Contemporary Theatre

700 Union Street  
Seattle, WA 98101  
Phone: 1-206-292-7660

Established in 1965 Seattle's premier theatre for contemporary plays. Bold, innovative productions on four unique stages in downtown Seattle.

# OOPSLA 2002 COMMITTEES

## Conference Committee

### Conference Chair:

Mamdouh Ibrahim, *IBM Global Services*

### Program Chair:

Satoshi Matsuoka, *Tokyo Institute of Technology*

### A/V Chair:

Tetsuhei Nakashima, *ITESM Campus Guadalajara*

### Communications Chair:

Jill Aden, *EDS*

### Demonstrations Chair:

Ken Bauer, *ITESM Campus Guadalajara*

### DesignFest® Chair:

Gail Harris, *Instantiated Software, Inc.*

### Doctoral Symposium Chair:

James Noble, *Victoria University of Wellington*

### Educators' Symposium Chair:

Helen Sharp, *The Open University*

### Internet Technology Chair:

Ken Bauer, *ITESM Campus Guadalajara*

### Objects and Web Services Chairs:

Brent Hailpern, *IBM Research*,  
Peri Tarr, *IBM Research*

### Onward! Chair:

Richard P. Gabriel, *Sun Microsystems*

### Panels Chair:

Brian Foote, *The Refactory, Inc.*

### Posters Chair:

Laurie Williams, *North Carolina State University*

### Practitioner Reports Chair:

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*

### Student Volunteer Chair:

Maria Elena (Helen) Chávez Echeagaray, *ITESM Campus Guadalajara*

### Technology Chair:

Peter Kriens, *aQute BV*

### Technical Papers Chair:

Satoshi Matsuoka, *Tokyo Institute of Technology*

### Treasurer:

Shail Arora, *Gradepoint, Inc.*

### Tutorials Chair:

Sherman Alpert, *IBM T. J. Watson Research Center*

### Workshop Chair:

Jutta Eckstein, *Objects in Action*

### OOPSLA 2003 Chair:

Ron Crocker, *Motorola*

### Exhibits:

Jim Johnson, *Meetings & More, Inc.*

### Liaison with ACM:

Donna Baglio, *ACM*

### Operations:

Nancy Wilson, Tammy Powers, & Colleen Puderbaugh  
*Meeting Strategies Worldwide*

### Registration:

Carole Mann, *Registration Systems Lab*

### CyberChair Operation and Support:

Richard van de Stadt, *Borbala Online Conference Services*

### Typesetting:

Erin Sunahara, *Sunahara Consulting*

### Webmaster:

Steve Watson, *Watson Sites*

## Program Committee

Mikio Aoyama, *Nanzan University*

David Bacon, *IBM Research*

Kent Beck, *Three Rivers Institute*

Elisa Bertino, *University Milan*

Gilad Bracha, *Sun Microsystems*

Frank Buschmann, *Siemens AG*

Craig Chambers, *University Washington*

Shigeru Chiba, *Tokyo Institute of Technology*

Jennifer Hamilton, *Microsoft*

Seongsoo Hong, *Seoul National University*

Tony Hosking, *Purdue University*

Gregor Kiczales, *U British Columbia*

Doug Lea, *SUNY Oswego*

Andrew Lumsdaine, *Indiana University*

Boris Magnusson, *University Lund*

Mira Mezini, *University Darmstadt*

Gail Murphy, *University British Columbia*

Oscar Nierstrasz, *University Bern*

James Noble, *Victoria University of Wellington*

Jens Palsberg, *Purdue University*

Martin Rinard, *MIT*

Marc Snir, *UIUC*

Guy L. Steele Jr., *Sun Microsystems*

Clemens Szyperski, *Microsoft Research*

John Vlissides, *IBM Research*

Allen Wirfs-Brock, *Instantiations*

## Tutorial Committee

Sherman Alpert, *IBM T.J. Watson Research Center (Chair)*

Stephen Fink, *IBM T.J. Watson Research Center*

Craig Larman, *Valtech*

Peri Tarr, *IBM T.J. Watson Research Center*

## Doctoral Symposium Committee

James Noble Ph.D., *Victoria University of Wellington (Chair)*  
 Prof. Alan Borning, *The University of Washington*  
 Dirk Riehle Ph.D.  
 Mira Mezini Ph.D., *University of Siegen and Northeastern University*  
 Mario Wolczko Ph.D., *Sun Microsystems Laboratories Inc.*

## Practitioner Reports Committee

Rebecca Wirfs-Brock, *Wirfs-Brock Associates (Chair)*  
 Lougie Anderson Ph.D., *Sabrix Inc.*  
 John Schwartz, *OHSU*  
 Jeffrey Wills Ph.D., *Valparaiso University*

## Educators' Symposium Committee

Helen Sharp, *The Open University (Chair)*  
 Gary Berosik, *West Group and University of St. Thomas*  
 Fintan Culwin, *London South Bank University*  
 Sally Fincher, *University of Kent at Canterbury*  
 Peter Long, *PWC Consulting*  
 Rick Mercer, *University of Arizona*  
 Ana Trevino, *ITESM Campus Guadalajara*  
 Eugene Wallingford, *University of Northern Iowa*

## Workshop Committee

Jutta Eckstein, *Objects in Action (Chair)*  
 Bruce Anderson, *IBM EMEA Component Technology Services*  
 John Daniels, *Syntropy Ltd.*  
 Steven D. Fraser, *Independent consultant*  
 Mary Lynn Manns, *University of North Carolina at Asheville*  
 Peter F. Sweeney, *IBM, T. J. Watson Research Center*  
 Markus Voelter  
 Laurie Williams, *North Carolina State University*

## Onward! Committee

Richard P. Gabriel, *Sun Microsystems (Chair)*  
 Geoff Cohen, *Cap Gemini Ernst & Young*  
 Pierre Cointe, *L'Ecole des Mines de Nantes*  
 Dave Thomas, *Bedarra Corporation and Carleton University*

## DesignFest Committee

Gail Harris, *Instantiated Software, Inc. (Chair)*  
 Torsten Layda, *SWX Swiss Exchange*  
 Daan Hoogland, *Snow B.V.*  
 James Heliotis, *Rochester Institute of Technology*  
 Lawrence Kucera, *The Boeing Company*  
 Michael Feathers, *Object Mentor Inc.*  
 Raul Gnaga, *SES/Astra*  
 Rob van den Berg, *Siemens VDO*

## Poster Committee

Laurie Williams, *North Carolina State University (Chair)*  
 Jutta Eckstein, *Objects in Action*  
 Eric Meijer, *Microsoft*  
 Granville Miller, *TogetherSoft*  
 Peri Tarr, *IBM T. J. Watson Research Center*

## ACM Student Research Competition Committee

Laurie Williams, *North Carolina State University (Chair)*  
 Richard P. Gabriel, *Sun Microsystems*  
 Brent Hailpern, *IBM T. J. Watson Research Center*  
 Erik Meijer, *Microsoft Corporation*  
 Helen Sharp, *The Open University*  
 Peri Tarr, *IBM T. J. Watson Research Center*  
 Rebecca Wirfs-Brock, *Wirfs-Brock Associates*

## Technology Committee:

Peter Kriens, *aQute BV (Chair)*  
 Niclas Nilsson, *Activa AB*

## OOPSLA Steering Committee

Linda Northrop, *Software Engineering Institute (Chair)*  
 Brent Hailpern, *IBM T. J. Watson Research Center (Past Chair)*  
 Mary Beth Rosson, *Virginia Tech*  
 Mamdouh Ibrahim, *IBM Global Services*  
 Ron Crocker, *Motorola*  
 John Vlissides, *IBM T. J. Watson Research Center*  
 Mike Burke, *IBM Research*  
 Tom Ball, *Microsoft Research*  
 Donna Baglio, *Director, ACM Office of SIG Services*